



La gestion de la qualité de service temps-réel dans les réseaux de capteurs sans fil

Khalid El Gholami

► To cite this version:

Khalid El Gholami. La gestion de la qualité de service temps-réel dans les réseaux de capteurs sans fil. Autre [cs.OH]. Université Blaise Pascal - Clermont-Ferrand II, 2014. Français. NNT : 2014CLF22537 . tel-01158110

HAL Id: tel-01158110

<https://theses.hal.science/tel-01158110>

Submitted on 29 May 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre D. U : 2537

EDSPIC: 682

UNIVERSITÉ BLAISE PASCAL - CLERMONT II

ÉCOLE DOCTORALE DES
SCIENCES POUR L'INGÉNIEUR DE CLERMONT-FERRAND

THÈSE EN COTUTELLE
AVEC
L'UNIVERSITÉ CHOUAIB DOUKKALI - EL JADIDA

Présentée par

EL GHOLAMI Khalid

Master Spécialisé en Réseaux et Télécommunications

Pour obtenir le grade de
DOCTEUR D'UNIVERSITÉ
Spécialité: INFORMATIQUE

La Gestion de la Qualité de Service Temps-Réel
dans les Réseaux de Capteurs Sans Fil

Soutenue publiquement le 20 décembre 2014 devant le jury :

Prof. Farouk Toumani	Université Blaise Pascal	Président
Prof. Fabrice Valois	INSA de Lyon	Rapporteur
Prof. Mohammed Boulmalf	Université Internationale de Rabat	Rapporteur
Prof. Rachid Hilal	Université Chouaib Doukkali	Rapporteur
Prof. Kun Mean HOU	Université Blaise Pascal	Directeur de thèse
Prof. Najib EL KAMOUN	Université Chouaib Doukkali	Directeur de thèse



Dédicace

^

À mes parents

A ma grand-mère

À ma sœur Soukaïna

À mes frères et leurs petites familles

À ma fiancée Ikram

À toute ma grande famille

À tous mes amis au Maroc, en France et en Asie

*À titre de reconnaissance, d'amour et d'affection, Je leur dédie ce
travail*

11//1/

Remerciements

J'exprime ma gratitude à Monsieur le Professeur kun-Mean HOU, responsable du groupe de recherche SMIR au sein du laboratoire LIMOS et directeur de ma thèse, d'avoir accepté de m'accueillir au sein de son équipe. Je le remercie pour l'excellence de son encadrement ainsi que pour la confiance et la grande autonomie qu'il m'a accordées tout en étant toujours disponible malgré ses nombreuses contraintes.

J'exprime ma reconnaissance à Monsieur le Professeur Najib ELKAMOUN, directeur du laboratoire STIC et directeur de ma thèse, de m'avoir encouragé à faire cette thèse et d'avoir accepté de diriger mes travaux de recherche. Je le remercie vivement pour ses conseils et ses orientations pédagogiques et scientifiques qui m'ont permis de surmonter les difficultés rencontrées au cours de cette période.

Je remercie Monsieur le Professeur Farouk Toumani, directeur du laboratoire LIMOS, d'avoir accepté de présider le jury de cette thèse. J'ai eu un grand bénéfice à profiter de ses conseils pertinents, de son savoir et de son expérience. J'exprime toute ma gratitude aux rapporteurs de ma thèse, Messieurs les Professeurs Fabrice Valois, Enseignant - Chercheur à l'INSA de Lyon, Mohammed Boulmalf, Enseignant - Chercheur et Doyen de la faculté ELIT à l'Université Internationale de Rabat et Rachid Hilal, Enseignant - Chercheur et vice-président de l'Université Chouaib Doukkali, qui ont accepté de juger ce travail de thèse et de partager avec nous leur savoir et expertise dans ce domaine de recherche. Je remercie plus particulièrement les membres de jury qui ont fait le déplacement jusqu'à El Jadida pour assister à ma soutenance de thèse.

Je tiens également à exprimer toute ma reconnaissance à Monsieur le Professeur Hassan Benabdillah pour son aide précieuse dans la rédaction de ce mémoire de thèse. Je remercie également Monsieur le Professeur Alain Quilliot, précédent directeur du laboratoire LIMOS de m'avoir accueilli dans son laboratoire et de m'accorder une grande confiance, et ce, tout au long de ma thèse.

Je veux également remercier tous les membres de l'équipe SMIR : permanent : Christophe de Vault et Jian-Jin Li, doctorants et post-doctorants : Muhammad Yusro, Yi-Bo Chen, Hong-Ling Shi, Xing Liu, Xunxing Diao (Peter), Bin Tian (Michael), Ding Hao et Zhou Peng, ainsi que les stagiaires. Et les membres du laboratoire STIC : Yassine Sabri, Fatima Lakrami, Abderrahim Maizate, Asmaa Kassid et les autres doctorants et professeurs qui ont tous largement contribué à l'excellente atmosphère de travail dans les deux laboratoires.

Je remercie chaleureusement le personnel de l'UBP et l'UCD qui m'ont facilité le travail au sein des deux établissements et qui ont mis à ma disposition les moyens pour mener mes travaux de recherche dans les meilleures conditions.

Je remercie toute ma famille pour leurs prières, leur soutien financier et leurs mots d'encouragement, depuis le début de cette thèse, mais également pendant tout mon cursus universitaire.

Le difficile, c'est ce qui peut être fait tout de suite.

L'impossible, c'est ce qui prend un peu plus de temps.

Georges Santayana

Table des matières

Dédicace	i
Remerciements.....	ii
Introduction Générale	9
CHAPITRE I : Généralités sur les RCSF et État de l'Art	12
1. Les réseaux de capteurs sans fil.....	13
1.1. Définition.....	13
1.2. Aperçu historique de l'évolution des réseaux de capteurs sans fil.....	13
1.3. Caractéristiques et contraintes d'un nœud capteur	15
1.3.1. Sources d'énergie	15
1.3.2. Mémoire.....	16
1.3.3. Processeurs embarqués.....	17
1.3.4. Module radio	17
1.4. Applications.....	18
1.4.1. Applications militaires	18
1.4.2. Applications environnementales	18
1.4.3. Applications médicales	19
1.4.4. Applications de traçabilité.....	19
1.4.5. Applications industrielles	20
1.4.6. Applications domotiques.....	21
1.5. Les stratégies de sommeil des nœuds dans les RCSF.....	22
1.5.1. On-demand Paging	22
1.5.2. Réveil synchrone.....	22
1.5.3. Réveil asynchrone.....	22
2. La qualité de service	23
2.1. Définition.....	23
2.2. Métriques	23
2.2.1. Délai	24
2.2.2. Gigue	24
2.2.3. Débit.....	25

2.2.4.	Élimination des paquets (drop)	25
2.3.	Garantie de la qualité de service dans les RCSF.....	25
2.3.1.	Solutions MAC	25
2.3.2.	Routage	26
2.3.3.	Traitement des données	26
2.3.4.	Cross-layer	26
3.	Le standard IEEE 802.15.4	27
3.1.	Couche physique PHY.....	27
3.2.	Sous-couche MAC.....	28
3.3.	Architecture de la pile protocolaire et les règles de communication entre les différentes couches.....	31
3.4.	Rapport cyclique (Duty Cycle) dans le standard IEEE 802.15.4.....	32
3.5.	Modèles analytiques pour le standard IEEE 802.15.4	32
4.	Travaux de recherche sur la QdS au niveau de la couche MAC	33
4.1.	Protocoles MAC avec contention	33
4.2.	Protocoles MAC sans contention	35
4.3.	Approches basées sur le choix des paramètres de la supertrame	37
4.4.	Cross Layer	38
4.5.	Le standard IEEE 802.15.4e.....	39
4.5.1.	Structure de la DSME multi-superframe	40
4.5.2.	Structure de la supertrame basée sur le LL-Beacons.....	40
5.	Conclusion.....	41
CHAPITRE II : Protocole FF-MAC, une Couche MAC pour une Meilleure QdS		42
1.	Contexte de l'étude	43
1.1.	Généralités	43
1.2.	Modèle Multi-source Multi-sink	43
2.	Problématique et limitations identifiées dans le standard IEEE 802.15.4.....	44
2.1.	Temps-réel	44
2.2.	Compromis énergie-délai.....	45
2.3.	Limitations identifiées dans le standard IEEE 802.15.4.....	46
3.	Amélioration du standard IEEE 802.15.4 pour une meilleure qualité de service temps réel.....	49
3.1.	Améliorations proposées	49
3.1.1.	Changement de l'ordre des périodes CAP et CFP	49
3.1.2.	Ajout de la nouvelle période D-CFP	50

3.1.3.	Trame PRTPA	51
3.2.	Généralités sur le protocole FF-MAC :	51
3.3.	Format de la trame PRTPA	53
3.3.1.	En-tête MAC :	54
3.3.2.	Charge de la trame PRTPA.....	55
3.3.3.	Pied de la trame	55
3.4.	Fonctionnement détaillé du protocole FF-MAC	56
3.5.	Protection de la période CAP :	60
3.6.	Exemple de déroulement du protocole FF-MAC.....	60
3.7.	Maintenance de la supertrame	62
3.8.	Tolérance aux pertes des paquets PRTPA	63
3.9.	Protocole FF-MAC et les autres approches.....	64
4.	Conclusion du chapitre.....	65
CHAPITRE III : Évaluation des Performances du Protocole FF-MAC par Simulation et Expérimentation		66
1.	Évaluation des performances par simulation	67
1.1.	Présentation du simulateur NS-2.....	67
1.2.	Modèle WPAN.....	68
1.3.	Avantages et limites de la simulation	68
1.4.	Paramètres de simulation.....	69
1.5.	Résultats obtenus	71
1.6.	Conclusion de la première partie.....	75
2.	Évaluation de performances expérimentale du protocole FF-MAC	76
2.1.	Importance de l'étude expérimentale dans la recherche appliquée	76
2.2.	Contraintes d'implémentation	76
2.3.	Détail technique et outils utilisés dans le prototypage	77
2.3.1.	Présentation de la plateforme de test	77
2.3.2.	Pile de protocoles Open Source « Atmel MAC Stack ».....	78
2.3.3.	Le logiciel JSensorsMonitor	80
2.4.	Implémentation du protocole FF-MAC.....	81
2.5.	Expérimentations.....	82
2.5.1.	Schéma expérimental	82
2.5.2.	Paramétrage des scénarios.....	83
2.5.3.	Validation de la nouvelle structure de la supertrame.	85

2.5.4.	Influence du changement des périodes sur le délai (nœuds vers le coordinateur)...	87
2.5.5.	Le relais rapide dans la même supertrame	88
2.5.6.	Aspect déterministe du protocole FF-MAC.....	93
2.5.7.	Impact du protocole FF-MAC sur les performances de l'algorithme CSMA-CA...	94
2.6.	Remarque importante.....	96
3.	Résumé des améliorations.....	96
4.	Conclusion du chapitre.....	97
CHAPITRE IV : Implémentation du Standard IEEE 802.15.4 sous le Simulateur NS-3.....		98
1.	Modèles existants	100
2.	Motivations	100
3.	Aperçus du Simulateur NS-3.....	101
4.	Conception et implémentation.....	102
4.1.	Architecture globale	102
4.2.	Composantes du modèle WPAN :.....	104
4.2.1.	Couche physique : WpanPhy.....	104
4.2.2.	Sous-couche MAC : WpanMac	104
4.2.3.	Sous-couche de convergence : WpanScs.....	106
4.2.4.	Gestionnaire du nœud : WpanDeviceManager.....	106
4.2.5.	WpanNetDevice	107
4.2.6.	Canal de transmission sans fil.....	107
4.2.7.	Modèle d'énergie	108
4.2.8.	Traces de simulation	109
4.2.9.	Assistants (Helpers).....	111
4.3.	Compatibilité avec le standard IEEE 802.15.4 et le modèle OSI	111
5.	Évaluation de performances.....	112
5.1.	Scénarios de test du modèle d'affaiblissement de propagation.....	113
5.2.	Mesure du délai pour les différentes couches MAC.....	117
5.3.	Mesure du débit pour les différentes couches physiques.....	120
6.	Conclusion du chapitre.....	124
Conclusion Générale et Perspectives		125
1.	Conclusion Générale.....	125
2.	Perspectives.....	126
Annexe A : Quelques observations expérimentales		129
Annexe B : Implémentation du mécanisme PRTPA		131

1. Dans le coordinateur.....	131
1.1. Création du descripteur des D-GTS	132
1.2. Préparation et transmission du paquet PRTPA	133
2. Dans les nœuds terminaux.....	138
Annexe C : Logiciel JSensorsMonitor	143
1. Architecture du logiciel.....	144
2. Fonctionnement du logiciel	144
Annexe D : Illustrations des équipements utilisés dans les expériences	146
Liste des publications.....	147
Bibliographie	148
Glossaire	156
Table des illustrations	161
Résumé:.....	163
Abstract:.....	164

Introduction Générale

L'utilisation des capteurs (sondes) dans divers domaines d'applications n'est pas une idée nouvelle. Cependant, les anciennes installations des capteurs étaient entièrement câblées. Cela pose plusieurs problèmes techniques comme l'inflexibilité, le coût élevé d'installation, la difficulté d'utilisation sur les appareils mobiles ou rotatifs, la difficulté du passage à l'échelle, etc. Ainsi, la technologie des réseaux de capteurs sans fil (RCSF) a été proposée comme une solution alternative aux capteurs câblés dans l'optique de profiter de ses avantages, notamment l'aspect sans fil. Néanmoins, les RCSF héritent aussi tous les problèmes de la communication sans fil, notamment ceux liés à l'accès au médium, comme la faible fiabilité de la communication sans fil, le problème du terminal caché et du terminal exposé.

Malgré sa ressemblance avec les autres réseaux sans fil, notamment les réseaux ad-hoc, les RCSF se distinguent par un ensemble de contraintes spécifiques. À titre d'exemple, les contraintes intrinsèques des nœuds RCSF se manifestent dans la pénurie des ressources d'énergie, la capacité limitée des mémoires et la faible puissance de calcul. Ces contraintes contribuent à augmenter la complexité de ces systèmes.

De plus, les réseaux de capteurs sans fil introduisent un changement de paradigme fondamental allant de la communication traditionnelle interhumaine à une communication inter-dispositifs autonomes. De plus, ces nœuds de RCSF à terme vont être déployés à large échelle et le nombre de nœuds sera plus important que celui du réseau d'internet actuel (p. ex. Internet des Objets). En effet, les équipements actuellement connectés sur internet sont des appareils qui nécessitent des utilisateurs pour les utiliser et les maintenir. C'est la raison pour laquelle leur nombre ne dépassera pas quelques milliards de nœuds. Par contre, les capteurs sans fil sont des dispositifs autonomes par définition. Ils ne nécessitent aucune intervention humaine après leur déploiement. En plus, dans la plupart des applications, ces nœuds sont déployés à très grande échelle pour collecter le maximum d'information et couvrir des zones géographiques plus larges avec une bonne résolution géographique. Cela implique une intégration d'un nombre pratiquement illimité de nouveaux nœuds. Ce nombre énorme de nouveaux nœuds qui rejoindront internet, dans le cadre des technologies d'Internet des objets et le Web des objets (Objets intelligents 'Smart Objects'), a poussé les chercheurs à adopter la technologie IPv6, vu l'énorme capacité d'adressage offert par ce protocole (adresse IP 128 octets au lieu de 32 octets). De plus,

l'organisation hiérarchique de l'IPv6 permet de simplifier les tâches de routage et de réduire considérablement la taille des tables de routage. L'adaptation de ce protocole pour les RCSF est assurée par la couche d'adaptation 6LoWPAN.

La qualité de service 'QoS' est l'un des concepts les plus importants dans les réseaux modernes. Elle est, de plus en plus, exigée à cause de l'intégration de plusieurs services ayant des besoins différents dans le même réseau. Dans un réseau de capteurs sans fil, les nœuds utilisent généralement une application unique. Néanmoins, les trafics en provenance des différents nœuds peuvent avoir des priorités différentes. Cette diversité de trafics dans le réseau nécessite, d'une part, un traitement différencié selon les besoins spécifiques de chaque type de trafic et d'autre part, la satisfaction des besoins des applications (p. ex. le délai). Dans le cadre des réseaux de capteurs sans fil, la problématique devient encore plus complexe à cause des contraintes intrinsèques des nœuds RCSF. La limitation des ressources (mémoire, processeur, énergie), la communication sans fil peu fiable, la densité importante et la nature distribuée des nœuds représentent les principaux enjeux pour le développement de tout protocole de communication. Bien évidemment, à cause de ces contraintes, la garantie de la QoS dans un réseau de capteurs sans fil pose des problèmes de recherche non triviaux. Ainsi, il est difficile d'appliquer directement les techniques utilisées dans les réseaux les plus proches au niveau de leurs architectures et mode de communication sans fil : les réseaux ad-hoc. Tous les mécanismes et protocoles doivent être adaptés et simplifiés pour qu'ils soient compatibles avec les contraintes de ressources d'un RCSF.

Comme les réseaux classiques, les approches proposées pour améliorer la QoS dans les RCSF se situent dans tous les niveaux du modèle OSI. La première couche qui influence les performances du réseau en entier se situe au niveau de liaison de données, plus particulièrement la sous-couche MAC 'Medium Access Control'. Les protocoles MAC sont responsables de la gestion d'accès au médium et, par conséquent, de prendre la décision pour transmettre une trame ou non. Ainsi, le choix de la méthode d'accès au médium à utiliser est déterminant si l'on souhaite garantir la QoS. L'un des protocoles MAC les plus utilisés dans le domaine des RCSF est le standard IEEE 802.15.4. Ce standard offre une implémentation très flexible, qui combine un accès aléatoire simple avec un accès déterministe permettant de répondre à des exigences en termes de bande passante et de délai de communication. Dans cette thèse, nous avons proposé un modèle de simulation de ce standard sous le nouveau simulateur NS-3. Malgré les avantages de ce standard, il présente quelques limitations dans ses mécanismes. Ainsi, un ensemble d'améliorations sont proposées dans la littérature pour pallier ses défauts. Notre travail s'inscrit

dans cette catégorie par la proposition une amélioration du standard afin corriger ses failles. Le protocole proposé permet de fournir un meilleur support de la qualité de service pour les applications temps-réel. La validation du protocole proposé a été réalisée par simulation et sur des plateformes de capteurs réelles.

Le mémoire de thèse s'organise en quatre chapitres: le premier chapitre est un état de l'art sur la technologie des réseaux de capteurs sans fil. Ce chapitre présente des généralités sur ce domaine de recherche en pleine évolution et aborde la problématique de la qualité de service dans les RCSF. De plus, un ensemble de protocoles proposés dans la littérature sont présentés.

Dans le second chapitre, nous proposons un nouveau protocole MAC à base du standard IEEE 802.15.4 permettant de garantir un accès déterministe de bout en bout en modifiant le format de la supertrame et en proposant une allocation dynamique des slots déterministes.

Le troisième chapitre dédié à la discussion des différents scénarios de tests effectués afin d'évaluer les performances de la méthode d'accès proposée. Dans cette validation, nous comparons le protocole FF-MAC avec le mécanisme des GTS proposé dans le standard IEEE 802.15.4. L'évaluation de performance est réalisée par simulation sous le simulateur NS-2 et par prototypage sur des nœuds RCSF réels en vue de valider le protocole proposé dans un environnement réaliste.

Le quatrième chapitre sera consacré à la conception, l'implémentation et la validation d'un modèle de simulation proposé pour le standard IEEE 802.15.4 sous le nouveau simulateur NS-3. Nous présenterons son architecture globale, ses différents composants en plus des nouveautés apportées par ce modèle par rapport au modèle de base existant dans le simulateur NS-2. Le modèle est testé à travers une série de scénarios afin d'évaluer ses performances et son bon fonctionnement.

CHAPITRE I : Généralités sur les RCSF et État de l'Art

Le premier chapitre présente les différents concepts et approches permettant de bien comprendre les contributions proposées dans cette thèse. Dans ce chapitre, nous présentons la technologie des réseaux de capteurs sans fil (RCSF), l'historique de son évolution, les contraintes et caractéristiques des nœuds d'un RCSF et les domaines d'application potentiels des RCSF. Ensuite, nous détaillerons le standard IEEE 802.15.4 qui fait l'objet de nos contributions. De plus, nous discutons un état de l'art des travaux de recherche dédiés aux protocoles de communication pour les RCSF, plus particulièrement, ceux qui ont proposé des améliorations à la couche MAC du standard IEEE 802.15.4 pour la fourniture d'une meilleur qualité de service 'QoS'.

1. Les réseaux de capteurs sans fil

1.1. Définition

Un réseau de capteurs sans fil (RCSF) est un réseau constitué d'un ensemble de nœuds capables d'acquérir des données (scalaires ou multimédias), de les traiter, puis de les transmettre à l'aide d'une interface de communication sans fil. Un nœud RCSF est composé d'un microcontrôleur, des mémoires pour stocker les données, des sondes, des convertisseurs analogique/numérique, un émetteur-récepteur radio, des contrôleurs qui lient ses composantes en plus d'une source d'énergie [1]. D'autres composantes peuvent être incorporées à cette architecture en fonction des applications cibles. Quant à la communication sans fil, elle est caractérisée par une portée très courte et un débit faible (LR-WPAN 'Low-Rate Wireless Personal Area Network'). Le trafic peut être généré d'une manière différente de celle des réseaux classiques. Parmi les modèles de trafic les plus utilisés dans les RCSF, on cite, le trafic périodique et le trafic événementiel [2]. Dans le premier modèle, le trafic est généré de manière périodique selon une fréquence définie par l'utilisateur. Cette dernière dépend de la nature de l'application et la contrainte d'énergie/délai. Dans le deuxième modèle, l'envoi des données est déclenché par des événements bien définis, comme à titre d'exemple, lorsque l'humidité du sol est insuffisante, quand la température est trop élevée, etc.

1.2. Aperçu historique de l'évolution des réseaux de capteurs sans fil

La technologie des RCSF a connu un progrès fulgurant grâce aux avancées technologiques dans le domaine des MEMS (Micro-Electro-Mechanical System), des circuits intégrés et des technologies de communication sans fil. Le RCSF peut s'intégrer à un spectre d'applications très large qui touche quasiment tous les domaines de la vie.

Comme la plupart des technologies avancées, les réseaux de capteurs sans fil ont vu le jour dans le domaine militaire et les applications industrielles importantes (p. ex. les réseaux de distribution électriques, le traitement des eaux usées et l'automatisation des usines « specialized factory automation »). Le système de surveillance sonore SOSUS « Sound Surveillance System » peut être considéré comme le premier réseau de capteurs qui présente des similitudes aux réseaux de capteurs sans fil modernes. Ce projet a été développé par l'armée américaine dans les années 1950 dans l'optique de détecter et suivre les sous-marins soviétiques à l'aide des capteurs acoustiques sous-marins [3].

Cependant, le développement des capteurs sans fil distribués n'a commencé qu'à partir de l'année 1980 avec le programme DSN « Distributed Sensor Network » initié par le DARPA [4]. Ensuite, avec la participation des grandes universités américaines dans ce projet (p. ex. Berkeley et MIT), cette technologie a trouvé une place dans la recherche académique et civile, ce qui a donné naissance à de nouvelles applications civiles.

Grâce à l'évolution extrêmement rapide dans ce domaine et l'énorme champ d'applications potentielles de cette technologie, les industriels n'ont pas tardé à rejoindre cette communauté, ce qui a donné une poussée vers l'avant en accélérant la production des dispositifs de nœuds RCSF tout en réduisant leur coût.

Ces avancées technologiques du côté matériel ont été accompagnées par une évolution sur le niveau logiciel avec l'apparition de nouveaux standards de communication sans fil proposés par les organismes de standardisation comme, à titre d'exemple ; l'IEEE (IEEE 802.15.4[5]), ISA [6] (ISA100.11a [7]) et l'alliance ZigBee [8] (ZigBee, ZigBee Pro, ZigBee IP et ZigBee RF4CE)[9]. De plus, un ensemble de systèmes d'exploitation dédiés aux RCSF ont vu le jour, comme TinyOS [10], Contiki [11], MIROS [12], etc.

Actuellement, les nœuds RCSF ont tendance à intégrer le monde d'internet pour profiter de ses atouts dans le cadre des technologies, Internet des Objets « IoT 'Internet of Things' » et Web des Objets « WoT 'Web of Things' » [13][14][15]. Selon Gartner¹, le nombre d'objet connecté va atteindre les 4,9 Milliards d'objets dans l'année 2015 avec une hausse de 30 % par rapport à l'année en cours (2014) et 25 milliards en 2020. Ils prévoient aussi que l'Internet des objets va créer une valeur de marché de 1.9 trillion de dollars u.s. au bout de l'année 2020 [17]. Ces statistiques et prévisions donnent une idée sur l'impact de ces technologies sur l'économie mondiale dans le futur.

Des applications identiques au système SOSUS que nous avons mentionné au début de cet aperçu existent toujours, mais pour des raisons plutôt pacifiques [18], comme la supervision de la vie sous-marine et des activités volcaniques, ce qui reflète l'évolution non seulement sur le plan technologique, mais aussi au niveau des types d'applications ciblées par les RCSF. Ces applications s'inscrivent actuellement en majeure partie dans la catégorie des applications scientifiques et civiles.

¹ Gartner Inc. Est une entreprise américaine de conseil et de recherche dans le domaine des technologies de l'information. Elle mène des recherches, fournit des services de consultation, tient à jour différentes statistiques et maintient un service de nouvelles spécialisées [16].

La figure 1 montre l'évolution des RCSF au cours du temps. Nous constatons tout d'abord que la nature des applications a passé du domaine militaire aux applications civiles en passant par les applications scientifiques et industrielles. Cette figure montre aussi une domination du marché par les applications grandes publiques qui touchent les différents aspects de la vie quotidienne. Le coût est réduit et la consommation d'énergie est atténuée de manière significative.

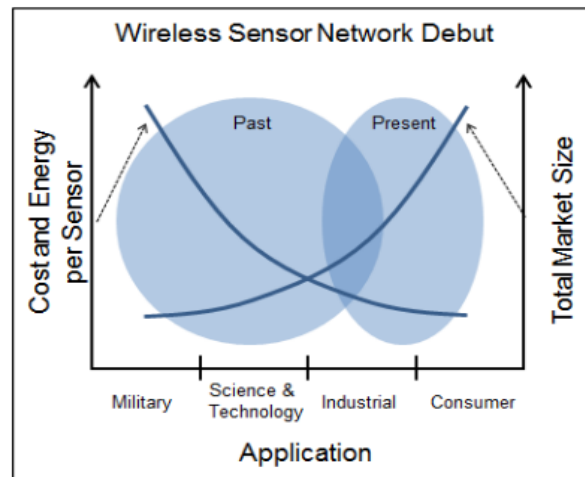


Figure 1 : Gain du marché des RCSFs avec la baisse des coûts des capteurs [3]

1.3. Caractéristiques et contraintes d'un nœud capteur

Les nœuds RCSF sont des systèmes embarqués qui souffrent de plusieurs contraintes intrinsèques. Ces dernières proviennent principalement des considérations économiques. En effet, le coût des nœuds RCSF doit être minimisé au maximum, car, dans plusieurs applications, le nombre de nœuds à déployer peut attendre des milliers de nœuds. De plus, la taille de ses nœuds RCSF peut aussi poser des problèmes technologiques ayant un impact sur leurs performances.

1.3.1. Sources d'énergie

L'un des principaux challenges qui font face à la conception de toute technologie liée aux RCSF et qui marquent ce type de réseaux est la pénurie des ressources d'énergie. En effet, l'alimentation des capteurs sans fil est généralement assurée par de simples piles de capacités limitées (p. ex. des batteries Lithium-ion [19]). L'utilisation d'une telle source d'énergie peut offrir de grands avantages, comme l'autonomie, le coût réduit et la flexibilité de déploiement. Néanmoins, cela peut donner naissance à des problématiques de recherches difficiles à résoudre vu que la capacité limitée de ces sources d'énergie. En pratique, le remplacement des piles n'est pas

toujours une opération facile, voir impossible dans certains cas de figure, comme le déploiement des nœuds RCSF dans des zones hostiles ou inaccessibles par les êtres humains. Ainsi, l'économie d'énergie est une priorité dans ce domaine. En effet, il est primordial d'assurer une gestion efficace de la consommation d'énergie afin de prolonger durée de vie des nœuds RCSF, sans pour autant dégrader leurs performances. Dans les premiers travaux de recherche dans le domaine des RCSF, les chercheurs ont fixé comme principal objectif la minimisation de la consommation d'énergie [20], notamment au niveau des protocoles de communication sachant que le module radio est souvent la composante qui consomme le plus d'énergie [21]. Cependant, Cette augmentation dans la durée de vie des nœuds RCSF est atteinte au détriment d'autres paramètres : le délai, la réactivité du réseau, etc. Notons que dans certaines applications, quelques nœuds RCSF peuvent être reliés à une source d'énergie permanente pour assurer la continuité de fonctionnement du réseau, notamment pour des nœuds spécifiques comme le puits ou encore le coordinateur du réseau personnel dans le cadre du standard IEEE 802.15.4. Des techniques de récolte d'énergie « Energy Harvesting » peuvent aussi être utilisées pour pallier ce problème [22] [23], mais avec un coût plus élevé.

Le Tableau 1 présente un exemple de dépenses énergétiques pour les différents modes de fonctionnement du module radio AT86RF230 [24]

Mode	Courants mesurés pour une puissance de sortie maximale
TRX_OFF	1,7 mA
BUSY_TX	17 mA
BUSY_RX	15,4 mA
RX_ON	15,7 mA
PLL_ON	8,5 mA

Tableau 1 : Modes de fonctionnement du module radio AT86RF230

1.3.2. Mémoire

La limitation de la capacité de mémoire des nœuds RCSF constitue l'une de ses importantes caractéristiques. Elle est constituée de deux types de mémoires, à savoir : la mémoire du programme, à partir de laquelle les instructions sont exécutées par le microcontrôleur (MCU) et la mémoire des données utilisée pour le stockage des données brutes « Raw data », échantillonnés par les sondes (données environnementales : température, humidité du sol, etc.).

Cette mémoire est très faible dans les nœuds RCSF actuels (Exemple de la plateforme iLive² : mémoire App./Boot « 128 Koctets » et la mémoire des données « 4 Koctets »).

1.3.3. Processeurs embarqués

Les microcontrôleurs actuels utilisés pour la réalisation d'un nœud RCSF sont très limités en termes de puissance de calcul. Plusieurs nœuds RCSF utilisés actuellement dans la recherche sont dotés de cœurs RISC 8-bits et d'une fréquence d'horloge de 16 MHz (plus d'exemples sont présentés dans le tableau 6, Chapitre III). Les tâches de calcul dans un réseau de capteurs sans fil peuvent inclure aussi bien les données échantillonnées localement par chaque nœud RCSF, que les données reçues des autres nœuds RCSF. En outre, le processeur est aussi une source consommatrice d'énergie et par conséquent, la conception de ces MCU doit prendre en considération l'efficacité d'énergie. Plusieurs techniques, telles que l'échelonnement dynamique de la tension DVS (Dynamic Voltage Scaling) et les modes de sommeil à énergie efficiente, sont incorporées en vue d'économiser de l'énergie. [25]

1.3.4. Module radio

Malgré la possibilité de couvrir une zone très large dans certaines applications, les réseaux de capteurs sans fil s'inscrivent dans la catégorie des réseaux personnels WPAN « Wireless Personal Area Network ». Ceci vient du fait que les échanges se font entre les nœuds RCSF dans un rayon de communication très court, allant de 10 à 100 mètres. La couverture peut être très large grâce à la communication multi-sauts assuré par des protocoles de routage dédiés. La bande passante offerte est aussi très faible : 10 à 250 kb/s selon le standard IEEE 802.15.4 (on peut attendre des débits jusqu'à 2 Mb/s en dehors du standard). Pour économiser l'énergie, les nœuds RCSF introduisent la notion du rapport cyclique « Duty Cycle » qui consiste en ce dont un nœud RCSF n'a pas besoin de rester actif tout le temps, il peut se mettre en mode sommeil et éteindre leurs émetteurs-récepteurs pour préserver de l'énergie.

Remarque : malgré toutes ces contraintes que nous avons présentées, la loi de Moore³ [26] nous rassure que les performances des nœuds RCSF seront meilleures dans le futur ce qui permettra de surmonter ses contraintes et promettra une ouverture sur de nouvelles applications.

² Plateforme développée par le groupe SMIR du laboratoire LIMOS UMR 6158 CNRS de l'Université Blaise Pascal

³Gordon Earle Moore, co-fondateur de la société Intel Corporation. Sa prédiction nommée « loi de Moore » est utilisée dans l'industrie des semi-conducteurs pour guider les planifications à long terme et définir des objectifs pour la recherche et le développement. Cette prédiction exprimée en 1965 dans « *Electronics Magazine* » puis réévaluée en

1.4. Applications

Comme nous l'avons mentionné dans l'introduction, les domaines d'application pour les réseaux de capteurs sans fil sont quasiment illimités [27]. La médecine, le domaine militaire, l'agriculture de précision, la domotique et l'automatisation industrielle sont des exemples, entre autres, de domaines d'applications pour les RCSF.

Ces applications peuvent se regrouper dans plusieurs classes selon leurs objectifs, les caractéristiques des trafics et les exigences relatives à la délivrance des données. Elles peuvent être aussi classifiées en fonction des types de nœuds RCSF utilisés [28], ou encore la topologie utilisée [29].

1.4.1. Applications militaires

Les applications militaires font partie des premières applications à utiliser des réseaux de capteurs sans fil. En effet, le domaine militaire a été la motivation principale du développement de cette technologie, vu que cette dernière permet de récupérer des informations importantes et très utiles sur les champs de bataille et les zones surveillées par l'armée. Les RCSF ont l'avantage de l'absence du câblage, et le coût réduit ce qui permet de déployer facilement un grand nombre de nœuds RCSF dans des zones hostiles. Plusieurs types de sondes sont utilisés pour acquérir les données (environnementales), à titre d'exemple ; le son, la lumière, l'énergie électromagnétique, la pression et les vapeurs chimiques, biologiques ou des explosives. Quelques exemples d'applications militaires sont présentés dans [28].

1.4.2. Applications environnementales

Les réseaux de capteurs sans fil sont, de plus en plus, utilisés dans le domaine de l'environnement. Ce domaine d'application est très intéressé par la collecte des informations environnementales avec une résolution spatio-temporelle très élevée. Ces besoins peuvent être satisfaits grâce à la facilité de déploiement et le coût réduit des capteurs sans fil. On trouve comme exemples de ces applications : la détection des incendies de forêt [30], l'enregistrement des activités volcaniques [31], le suivi des espèces en voie de disparition [32], etc. [33].

1975 s'est révélée étonnamment exacte. Elle postule que le nombre de transistors des microprocesseurs sur une puce de silicium double tous les deux ans.

Un travail réalisé par le centre CSIRO en Australie [34] d'une part, couvre un ensemble complet d'applications environnementales et d'agricultures, et d'autre part, présente les applications des RCSF environnementales dans le passé et les futures opportunités.

1.4.3. Applications médicales

La médecine fait partie des domaines qui utilisent les sondes depuis très longtemps pour mesurer les signaux vitaux des patients. Cependant, avec l'évolution des RCSF, l'élimination des câbles et l'intégration des capteurs sur internet donneront naissance à de nouvelles applications qui faciliteront la vie aussi bien aux médecins qu'aux patients. Par exemple, grâce à l'utilisation des nœuds RCSF connectés sur internet, le médecin peut suivre l'état de santé de ces patients alors que ces derniers mènent leurs vies de manière naturelle. L'utilisation, par exemple, des microsondes qui pourront être ingérées ou implantées sous la peau éliminera toute gêne des patients [35] et aura, par conséquent, un impact positif sur l'état psychique des malades, notamment ceux qui souffrent de maladies chroniques. Ainsi, toute anomalie enregistrée sur l'état de santé d'un patient sera signalée au médecin pour qu'il puisse intervenir [36]. Cette application participera aussi à réduire la charge dans les hôpitaux, et les coûts de déplacement et des visites médicales périodiques des patients. La canne intelligente est un autre exemple d'application des RCSF. Cette application aide les aveugles ou les personnes ayant des troubles visuels pour se déplacer en toute sécurité et de manière autonome dans les endroits publics [37].

Mais, vu que ce domaine d'application touche la santé et la vie des gens, il est primordial d'avoir des nœuds RCSF de très haute fiabilité, en plus d'une maturité des technologies utilisées, pour assurer le bon fonctionnement du système.

1.4.4. Applications de traçabilité

Les applications de traçabilité « target tracking » sont très importantes dans plusieurs domaines. Elles permettent de faire, par exemple, le suivi des véhicules de transport de biens, la localisation des victimes de catastrophes naturelles comme les avalanches [38] et dans les zones dangereuses comme les mines [39], le suivi des animaux pour les étudier, la traçabilité de déplacement des ennemis dans les champs de bataille, etc. Une taxonomie présentée dans [40] classe les algorithmes de traçabilité en se basant sur la l'architecture des réseaux dans deux grandes catégories : hiérarchique et point à point. Ces deux catégories regroupent d'autres sous-catégories comme ils sont illustrés dans la figure 2.

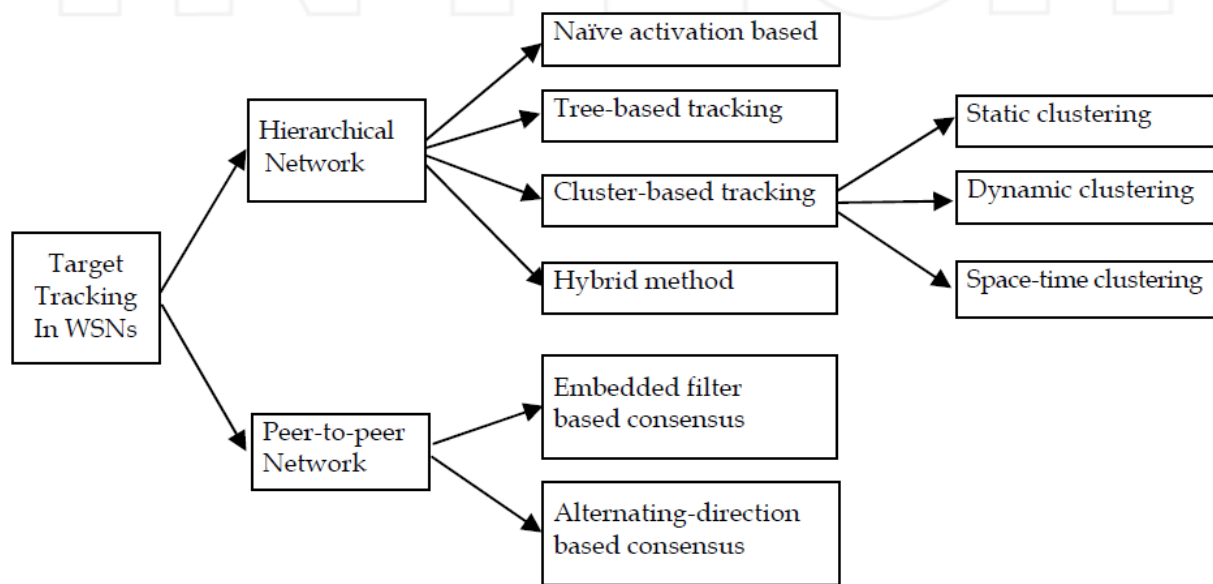


Figure 2 : Taxonomie des algorithmes de traçabilité dans les RCSF [40]

1.4.5. Applications industrielles

Le contrôle et la surveillance sont les tâches les plus importantes pour les RCSF dans les applications industrielles. Selon [41], les applications industrielles sont caractérisées par un ensemble de besoins essentiels qui se résument dans : l'économie d'énergie, la qualité de service, l'auto-configuration, la mobilité et la sécurité.

Selon [42][43], les applications des RCSF dans un environnement industriel « IWSN » peuvent être classifiées en trois catégories principales en fonction des besoins spécifiques de la production industrielle, à savoir : (i) l'acquisition des données environnementales, qui constitue le type d'applications principal dans ce domaine. L'objectif de ce genre d'applications est de collecter des informations diverses sur l'environnement en question. Les RCSF apportent l'avantage de la communication sans fil qui élimine tout inconvénient causé par les câbles. (ii) La supervision de la condition : ces applications sont très importantes dans le domaine industriel, car ils permettent, par exemple, de superviser l'état d'amortissement d'un équipement en vue de prédire les défauts et anticiper leur réparation ou leur remplacement. (iii) L'automatisation des processus, quant à elle, se charge de la surveillance des processus et des services. L'objectif du premier est d'évaluer et d'améliorer chaque étape du cycle en entier : la production, la distribution et la consommation des biens. Le second se charge de l'évaluation de la qualité de la mise à disposition d'un service spécifique à un utilisateur final.

1.4.6. Applications domotiques

La domotique est l'un des principaux domaines d'application pour les réseaux de capteurs sans fil dans le futur. En effet, grâce aux avancées technologiques dans les appareils utilisés à domicile (TV, climatiseur ...), qui deviennent, de plus en plus, connectés, l'utilisation des capteurs sans fil peut servir dans plusieurs applications liées à la sécurité, l'optimisation de la consommation d'énergie, l'automatisation des tâches, surveillance des enfants, etc. Un exemple de ces applications est illustré dans la figure 3.

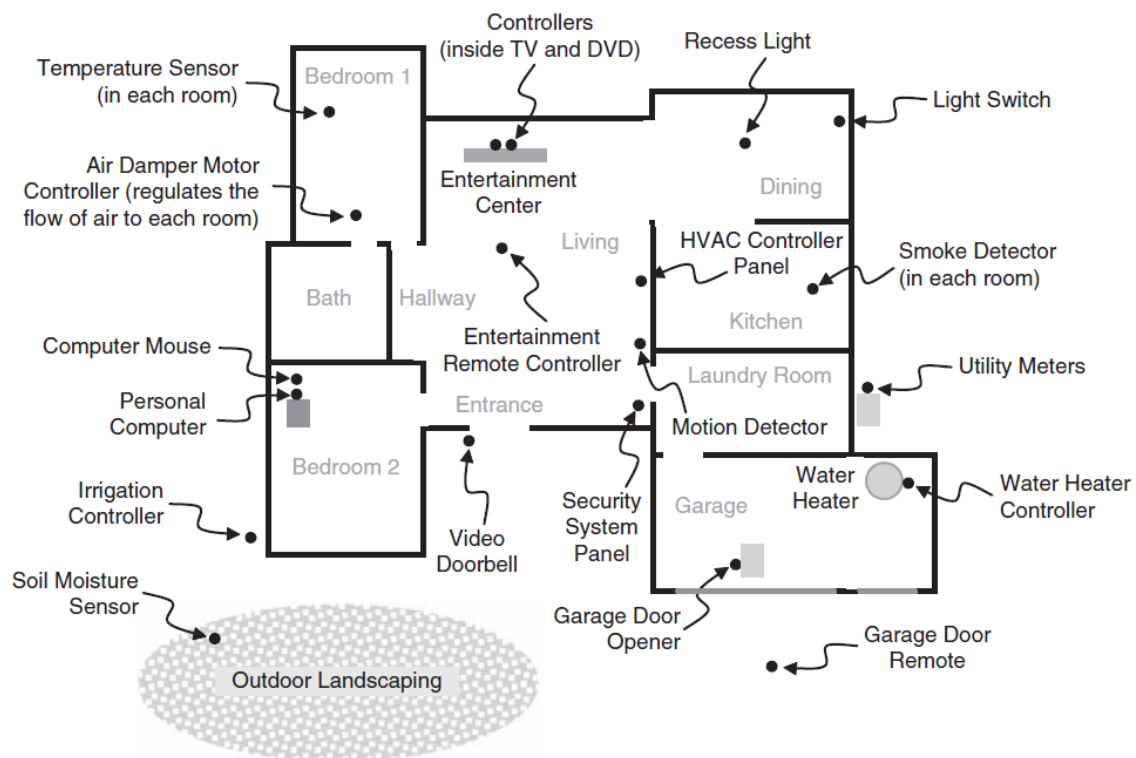


Figure 3 : Exemple de réseau de capteurs dans un bâtiment résidentiel typique [44]

Ces applications ont pour objectif de faciliter la vie des gens et de les aider à avoir un contrôle total de leurs habitats tout en optimisant l'utilisation des différentes ressources comme l'électricité, l'eau et le gaz par exemple.

Remarque : dans le cadre de cette thèse, on s'intéresse aux applications qui demandent un certain niveau de QdS en termes de délai de communication. Néanmoins, dans l'ensemble de ces applications, on peut trouver des scénarios où le besoin de la QdS doit être envisagé. Ainsi, nous avons défini le contexte de notre étude en fonction du modèle de communication et de la nature des trafics qui circulent dans le réseau, au lieu de préciser une application spécifique.

1.5. Les stratégies de sommeil des nœuds dans les RCSF

En vue de réduire la consommation d'énergie, les chercheurs ont proposé plusieurs techniques permettant de mettre le nœud RCSF au repos lorsqu'il n'a pas besoin d'échanger des données avec ses voisins [45]. On peut distinguer trois catégories :

1.5.1. On-demand Paging

L'idée de base de cette stratégie a été proposée dans [46][47]. Elle consiste à éteindre complètement l'émetteur-récepteur lorsqu'un nœud RCSF se trouve dans le mode dormant. Le nœud RCSF utilise un dispositif séparé, qui consomme très peu d'énergie, pour se réveiller. Ce dispositif de « paging » reste toujours actif et utilise les messages reçus pour activer le nœud RCSF.

D'autres chercheurs proposent l'utilisation d'autres types de technologie radio dans ce que l'on appelle « wake-on-wireless ». L'objectif est de garder la composante radio inactive quand il n'y a pas de données à échanger. Ceci a été proposé pour la technologie Wi-Fi dans [48] et pour le Bluetooth dans [49]. La technologie RFID peut être aussi utilisée pour cet effet [50]. Cependant, la RFID consomme une énergie considérable ce qui la rend moins attractive que les autres solutions.

1.5.2. Réveil synchrone

Le principe du réveil synchronisé est de permettre aux nœuds RCSF de basculer entre l'état de sommeil et l'état actif de manière entièrement coordonnée. Ainsi, tous les nœuds RCSF du réseau savent exactement quand le réseau se met en mode sommeil et quand la période active commence. Cette synchronisation permet une meilleure gestion des ressources d'énergie. Les deux standards IEEE 802.11 [51] et IEEE 802.15.4 dispose de cette option pour pouvoir économiser l'énergie.

1.5.3. Réveil asynchrone

Dans ce mode de réveil, aucune coordination n'est envisagée. Les nœuds n'ont aucune connaissance a priori de l'ordonnancement global ou local des nœuds RCSF des réseaux. Ainsi, ce mode est supposé être moins consommateur d'énergie grâce à l'absence de la surcharge nécessaire pour la gestion et la maintenance de la synchronisation (Beacon par exemple). Il se base essentiellement sur l'échantillonnage de préambule à une fréquence bien définie pour détecter les communications en cours. Le réseau peut aussi utiliser des nœuds ayant moins de

contraintes d'énergie en tant que points de relai. Néanmoins, étant donné que les nœuds RCSF ne sont pas synchronisés entre eux, le contact d'un nœud RCSF dormant pourra être très coûteux en termes d'énergie [52].

Plusieurs protocoles MAC ont été proposés en se basant sur le mécanisme du Duty Cycle dans les deux versions (synchrone et asynchrone) pour prolonger la durée de vie des nœuds RCSF [53].

2. La qualité de service

2.1. Définition

La qualité de service « QoS », ou QoS « Quality of Service », est définie par l'UIT [54] dans la recommandation E.800 comme étant « *l'effet procuré par la qualité de fonctionnement d'un service qui détermine le degré de satisfaction de l'utilisateur* ». Ainsi, elle correspond à la performance globale d'un réseau, qu'il soit téléphonique ou informatique, perçue par les utilisateurs du réseau. Ce concept est très important pour le transport des trafics ayant des besoins spécifiques. En effet, dans les réseaux à ressources partagées (p. ex. Internet, RCSF ...), il est important d'offrir un service différencié permettant de gérer les ressources de manière optimisée en favorisant les trafics ayant des besoins spéciaux par rapport aux autres trafics qui tolèrent un niveau de qualité de service minimal. Ainsi, les mécanismes de la QoS nous permettent d'intégrer de nouvelles applications dans des réseaux qui ne sont pas dimensionnés particulièrement pour elles, tout en gardant un fonctionnement acceptable pour les applications de base.

Selon [55], la qualité de service est affectée par plusieurs facteurs qui peuvent être répartis en deux catégories : le facteur humain comme à titre d'exemple, la stabilité du service, le délai et la disponibilité. Ces facteurs sont subjectifs, car ils sont liés à la satisfaction de l'utilisateur final qui peut être différente d'une personne à l'autre ou pour la même personne dans des situations variées. Les facteurs techniques, quant à eux, sont des facteurs mesurables et, par conséquent, permettent une évaluation objective et plus précise. Ces facteurs techniques peuvent être illustrés dans la fiabilité, l'extensibilité « Scalability », l'efficacité, etc.

2.2. Métriques

Afin de quantifier la qualité de service, un ensemble de critères appelés "métriques de performance" sont considérés. Ces métriques permettent d'évaluer la QoS d'une façon plus

objective au lieu de se contenter de la satisfaction des gens. Néanmoins, les valeurs de référence pour ces métriques représentent les valeurs permettant de satisfaire les besoins des utilisateurs des services. Par exemple, selon la norme G.114 de l'ITU-T, le délai global recommandé pour les communications voix dans un seul sens est de 150 ms et d'un maximum de 250 ms. Au-delà de cette dernière valeur, la communication interactive devient lente et gênante pour les utilisateurs. Parmi les métriques les plus utilisées, on trouve :

2.2.1. Délai

Le délai de communication est une métrique très importante, car la plupart des applications recommandent une communication rapide. Cependant, les exigences en termes de délai changent d'une application à l'autre, ou même entre les différents messages de la même application. On distingue deux types d'applications selon leur degré de tolérance au délai : (i) les applications tolérantes en délai, appelé SRT « Soft Real-Time », qui peuvent tolérer un délai maximal moyen et (ii) les applications qui ont des exigences strictes en termes de délai maximal (appelé HRT « Hard Real-Time »). Ces deux types sont discutés dans la section 2.1 (Chapitre II).

Le délai provient généralement de deux sources principales. (i) Le retard d'accès au médium qui dépend fortement de la densité des nœuds RCSF, de la bande passante offerte par la couche physique et de la méthode d'accès au canal utilisée. Ce délai peut être maîtrisé dans le cas des méthodes d'accès déterministes. (ii) Le deuxième provient du temps de traitement des paquets à l'intérieur des nœuds intermédiaires (routeurs), plus particulièrement dans les files d'attente. Ce type de délai est généralement appelé latence. Il dépend plutôt de la charge du trafic, de la bande passante et des mécanismes d'ordonnancement et de gestion des files d'attente utilisées. D'autres sources de délai peuvent être négligeables comme, à titre d'exemple, le délai de propagation vu que la distance entre les nœuds est très limitée dans le domaine des réseaux de capteurs sans fil.

2.2.2. Gigue

La gigue, par contre, est la variation du délai des paquets reçus au fil du temps. Cette gigue vient du fait que les conditions réseau ne sont pas toujours stables et peuvent varier d'un instant à l'autre. Ainsi, cette variation peut affecter la qualité de service dans certaines applications comme les applications interactives. Pour palier le problème de variation de délai, on utilise une mémoire tampon au niveau du récepteur afin d'ajuster le délai pour qu'il soit constant. Cependant, si la gigue d'un paquet est très large, ce dernier risque d'être supprimé. Lorsque le

nombre de paquets supprimés est important, cela se traduira à problèmes de reconstitution des données, ce qui affectera la qualité de service (p. ex. détérioration de la qualité du son, de l'image ...)

2.2.3. Débit

Étant donné que la bande passante définit la capacité de la couche physique en termes de quantité de données qu'elle est capable de transmettre par unité de temps (elle prend une valeur fixe), le débit représente l'occupation réelle de la bande passante. En effet, la bande passante correspond au maximum théorique du débit. Ce dernier peut être affecté par plusieurs facteurs comme, entre autres, la densité des nœuds RCSF, la fiabilité du médium de transmission sans fil et le type de protocole utilisé pour la gestion de l'accès au médium (MAC).

2.2.4. Élimination des paquets (drop)

Les nœuds peuvent procéder à une élimination des paquets pour plusieurs raisons. Par exemple, ils peuvent rejeter les paquets corrompus à cause des problèmes de bruit et des interférences, surtout dans les communications sans fil. La récupération des paquets grâce aux codes correcteurs peut être très coûteuse en matière de puissance de calcul, notamment pour les nœuds RCSF. C'est la raison pour laquelle il est préférable de retransmettre les paquets erronés au lieu de les corriger. Cependant, la retransmission provoque des délais supplémentaires et une consommation d'énergie additionnelle. Un autre cas du rejet des paquets se manifeste lorsque les files d'attente des nœuds intermédiaires (routeurs) sont remplies. Ce dernier cas peut être très présent dans les RCSF, car la taille de mémoire des nœuds RCSF est très limitée. Néanmoins, dans la plupart des applications, la charge du trafic est relativement faible ce qui limitera les dégâts causés par ce phénomène.

2.3. Garantie de la qualité de service dans les RCSF

La garantie de la qualité de service peut se faire selon plusieurs approches et les mécanismes utilisés peuvent exister dans tous les niveaux du modèle OSI.

2.3.1. Solutions MAC

Les protocoles MAC « Medium Access Control » sont responsables de la gestion de l'accès au canal sans fil partagé. Cette gestion se fait selon plusieurs méthodes en fonction des besoins des applications, à savoir ; les méthodes avec contention, les méthodes sans contention et

les méthodes d'accès hybrides. Plus de détail sur les travaux de recherche des couches MAC en relation avec nos propositions est discuté dans la section 3.5.

2.3.2. Routage

Même si la couche MAC arrive à garantir la qualité de service localement (un saut), la qualité de service de bout en bout ne peut pas être assurée dans un environnement multi-sauts s'il n'y a pas de garantie au niveau de la couche réseau. Les protocoles de routage participent à la garantie de la qualité de service par le choix du chemin optimal en vue de transmettre les paquets tout en respectant les besoins de l'application à travers plusieurs techniques de routage [56]. L'objectif du protocole de routage dans ce cas est de trouver le chemin le plus court, mais qui offre une garantie des besoins requis par l'application cible. Les protocoles de routage existant offrent aussi une garantie de QoS soit probabiliste ou déterministe. On note qu'il existe des protocoles MAC conçus pour la communication multisaut, notamment la topologie en arbre de clusters (cluster tree), comme les protocoles DMAC [57] et DB-MAC [58]. Ce type de protocoles MAC peut de garantir la QoS de bout en bout à ce niveau du modèle OSI (Open System Interconnection) de l'ISO (International Organization for Standardization).

2.3.3. Traitement des données

L'agrégation des données permet d'améliorer l'efficacité en énergie, de réduire la charge du trafic global, notamment dans le « Backbone » du réseau et les points les plus proches de la destination, et par conséquent, d'améliorer la qualité de service. [59].

Il existe plusieurs techniques utilisées pour l'agrégation des données comme les approches basées sur les clusters [60], les approches basées sur l'arbre d'agrégation [61], l'agrégation basée sur le traitement distribué des données à l'intérieur du réseau « In-Network » [62] et l'agrégation centralisée qui se fait dans un serveur à l'extérieur du réseau « out-of-network ».

2.3.4. Cross-layer

Dans les modèles de couches (OSI et TCP/IP par exemple), le rôle de chaque couche est d'offrir des services à la couche supérieure adjacente. L'intérêt de ce modèle de communication est de garder une transparence pour éviter à ce que les couches supérieures connaissent le détail des autres couches inférieures. Toutes les communications doivent se faire à travers des interfaces prédéfinies par le protocole utilisé dans chaque couche. Ceci a donné comme résultats, une facilité de standardisation des protocoles de communication, une aisance dans le développement de nouveaux protocoles de toutes les couches, en plus de l'interopérabilité entre différents

réseaux et équipements. Malgré tous ces avantages, ce modèle n'est pas très flexible, car les communications s'effectuent de manière stricte et prédéfinie. Cela peut mener dans certains cas à une utilisation moins efficace des ressources du réseau, et par conséquent, à une dégradation de la qualité de service, notamment dans les environnements dynamiques. Ceci a motivé des chercheurs à adopter le modèle de cross-layer [63]. Le concept du cross-layer est centré sur l'idée d'optimiser le contrôle et l'échange des informations entre deux ou plusieurs couches de la pile protocolaire utilisée à travers un agent ou un plan de gestion partagé par les différentes couches. L'objectif est d'améliorer les performances en exploitant cette interaction entre les protocoles des différents niveaux du modèle OSI.

3. Le standard IEEE 802.15.4

Le standard IEEE 802.15.4 est l'un des principaux protocoles candidats pour les réseaux de capteurs sans fil, Internet des objets (IoT : Internet of Things), web des objets (WoT : Web of Things). Ce standard spécifie les deux couches basses du modèle OSI, en l'occurrence, la couche physique (PHY) et la sous-couche MAC. Il est optimisé pour les réseaux à faible débit en respectant les limitations de ressources (principalement, l'énergie). Les protocoles proposés dans le standard (c.-à-d. MAC et PHY) sont très adaptés à ce type de réseaux, car ils offrent des fonctionnalités qui prennent en considération les contraintes des nœuds RCSF.

3.1. Couche physique PHY

La couche physique « PHY » est la partie responsable de la transmission du flux binaire, constituant la trame en attente de transmission, au canal de transmission sans fil à travers l'émetteur-récepteur radio. Elle est aussi responsable de la réception et le décodage des signaux reçus et leurs délivrances à la sous-couche MAC. En outre, cette couche contrôle l'état de l'émetteur-récepteur (c.-à-d. envoi/réception et activation/désactivation), la sélection du canal de fréquences de communication, en plus de l'offre des services demandés par la couche supérieure adjacente, en l'occurrence la couche MAC (p. ex. la détection d'énergie, la vérification de la disponibilité du canal, la mesure de la qualité du lien pour les paquets reçus, etc.).

La couche physique du standard IEEE 802.15.4 peut opérer dans l'une des trois bandes radio ISM (Industrial, Scientific and Medical) suivantes : 868 MHz, 915 MHz et 2.4 GHz avec des débits faibles allant de 10 kb/s à 250 kbps. Les différentes caractéristiques des couches physiques proposées dans ce standard sont présentées dans le tableau 2.

PHY (MHz)	Frequency band (MHz)	Spreading parameters		Data parameters
		Modulation	Bit Rate (kb/s)	Symbol rate (Ksymbol/s)
868/915	868–868.6	BPSK	20	20
	902–928	BPSK	40	40
868/915 (optional)	868–868.6	ASK	250	12.5
	902–928	ASK	250	50
868/915 (optional)	868–868.6	O-QPSK	100	25
	902–928	O-QPSK	250	62.5
2450	2400–2483.5	O-QPSK	250	62.5

Tableau 2 : Bandes de fréquence et débits binaires et de symboles disponibles [5]

3.2. Sous-couche MAC

Le standard propose deux protocoles MAC alternatifs, le premier fonctionne en mode avec beacon activé « beacon-enabled » et le deuxième en mode avec beacon non activé « non beacon-enabled ». Dans ce dernier mode, l'accès au canal est géré par l'algorithme CSMA-CA « Carrier Sense Multiple Access with Collision Avoidance » non slotté pour éviter les collisions. Cet algorithme se distingue de la version CSMA-CA utilisée dans la technologie Wi-Fi (c.-à-d. IEEE 802.11) par la suppression du mécanisme RTS/CTS, vu que la taille d'une trame IEEE 802.15.4 est très petite. En effet, la charge de la couche physique ne doit pas dépasser 127 octets, ce qui élimine tout intérêt d'utiliser les paquets RTS et CTS. Dans ce mode MAC, la transmission périodique du beacon n'est pas activée et sa transmission n'est faite qu'à l'issue d'une demande d'association initiée par les nœuds RCSF. Ainsi, la synchronisation n'est pas requise et aucun mécanisme de qualité de service n'est fourni. Ces caractéristiques rendent ce mode plus adapté aux applications n'ayant pas de besoins spécifiques de qualité de service. par ailleurs, le mode avec beacon activé utilise une structure temporelle, appelé supertrame, pour gérer l'accès au médium partagé. Cette supertrame, illustrée dans la figure 4, est constituée de deux sections principales. La première est la section active, utilisée pour l'échange des trames entre les différents nœuds du réseau. Cette section peut être divisée en deux périodes ; la période avec contention « CAP 'Contention Access Period' » où les nœuds du réseau utilisent l'algorithme CSMA-CA dans sa version slottée pour gérer l'accès au canal et la période sans contention « CFP 'Contention Free Period' » qui est optionnelle. Dans cette dernière, le canal de transmission est réservé de manière exclusive pour certains nœuds ayant des besoins spécifiques en termes de débit ou de délai de communication. Ces nœuds utilisent un accès multiple à répartition dans le temps, TDMA « Time Division Multiple Access », sous forme de slots temporels adjacents appelés GTS « Guaranteed Time Slot ». Ces derniers sont limités à un nombre de 7 GTS dont

chacun peut être composé d'un ou plusieurs slots temporels. La transmission des trames de données dans les GTS est unidirectionnelle et s'effectue seulement entre le nœud propriétaire du GTS et son coordinateur. La deuxième section est la période de sommeil qui est aussi optionnelle et dans laquelle aucune transmission n'est autorisée puisque les nœuds du réseau doivent éteindre leurs modules radio pour économiser de l'énergie. Ce mode de fonctionnement est bien adapté aux applications ciblées par ce standard comme les réseaux de capteurs sans fil, car selon le standard, le rapport cyclique prévu pour des applications typiques est inférieur à 1 %.

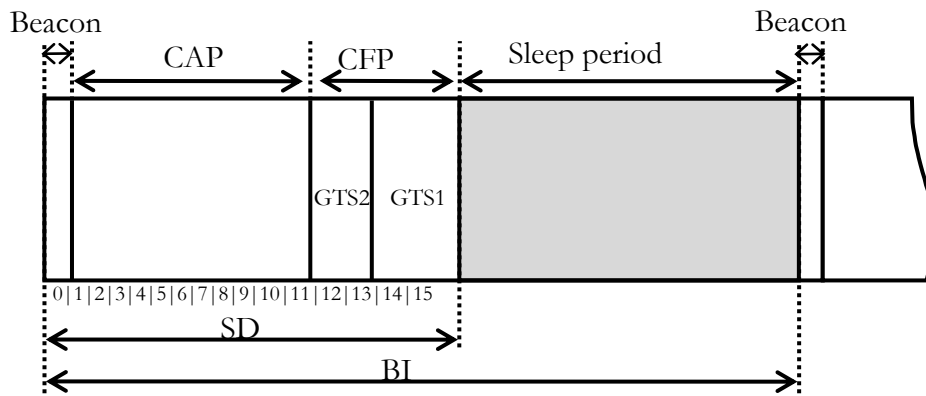


Figure 4 : Structure de la supertrame avec les périodes optionnelles

Dans le mode beacon-enabled, tout le PAN « Personal Area Network » est géré par un nœud appelé coordinateur du PAN. Ce dernier est responsable de l'envoi périodique du beacon au début de chaque supertrame. La trame de beacon est le moyen utilisé par le coordinateur pour synchroniser les nœuds associés à la supertrame, décrire la structure de cette dernière et identifier le PAN. Le beacon peut contenir des informations supplémentaires sur les adresses des nœuds ayant des trames en attente chez le coordinateur, en plus de la configuration des GTS, le cas échéant. La répartition de la supertrame est basée essentiellement sur les paramètres BO « Beacon Order » et SO « Superframe Order ». Le paramètre « Final CAP » est aussi utilisé pour marquer la fin de la période CAP en présence des GTS. Ainsi, les nœuds du réseau peuvent utiliser ces informations pour identifier la durée de la section active, la durée de la période CAP, la durée de la section de sommeil et la durée des slots. Les calculs se font par les équations suivantes :

$$BI = aBaseSuperframeDuration \times 2^{BO}(\text{symboles}) \quad (I)$$

$$SD = aBaseSuperframeDuration \times 2^{SO}(\text{symboles}) \quad (II)$$

$$sd = aBaseSlotDuration \times 2^{SO} = \frac{SD}{16} (\text{symboles}) \quad (III)$$

$$\text{tel que : } 0 \leq SO \leq BO \leq 14$$

Où *aBaseSuperframeDuration* et *aBaseSlotDuration* sont deux constantes prédéfinies par le standard et sont égales à 960 et 60 symboles respectivement. Elles représentent la taille minimale de la supertrame et celle des slots respectivement. Chaque symbole correspond à un nombre de bits en fonction de du type de couche physique sélectionnée (p. ex. un seul symbole correspond à 4 bits dans la couche 2.4 GHz PHY). BI « Beacon Interval » est la taille totale de la supertrame qui sépare la transmission de deux trames beacons successives. Finalement, la durée de la supertrame « SD » est en effet, la durée de la période active de la supertrame. Elle est constituée de 16 slots de tailles identiques et égales à *sd* « slot duration ».

Un mode supplémentaire, appelé le mode sans beacon « beaconless », qui n'est pas défini dans le standard, peut être utilisé comme une alternative dans les réseaux qui ont besoin d'une couche MAC très simples, plus particulièrement dans les réseaux maillés « mesh ». En effet, ce mode n'utilise que la couche physique définie dans le standard, en plus du mécanisme CSMA-CA comme protocole de la couche MAC. Ainsi, aucune association n'est supportée et les nœuds du réseau ont des rôles équivalents (au niveau MAC). Dans ce cas de figure, les nœuds utilisent l'adressage étendu au format 64-bits, car il n'y a plus de coordinateur du PAN pour distribuer les adresses courtes (c.-à-d. le format 16-bits).

Résumé des fonctionnalités de la couche MAC :

- Génération des beacons (dans le cas du coordinateur).
- Fourniture des services d'association et de dissociation du PAN.
- Synchronisation des nœuds aux beacons (dans le mode « beacon-enabled »).
- Utilisation de l'algorithme CSMA-CA et la gestion des GTS pour accéder au canal.
- Assurance de la fiabilité des liens entre deux nœuds dans le niveau MAC.
- Fourniture des mécanismes de sécurité dans le niveau MAC.

3.3. Architecture de la pile protocolaire et les règles de communication entre les différentes couches

Comme il est indiqué sur la figure 5, les échanges entre la sous-couche MAC et la couche physique doivent respecter les règles du modèle OSI, notamment le concept d'abstraction des couches. En effet, chaque couche communique avec l'autre exclusivement à travers des interfaces standardisées appelées des points d'accès aux services « SAP » dans lesquels chaque couche fournit des services à sa couche supérieure. Chaque service est accessible via quatre primitives [64] au maximum, à savoir : demande, indication, réponse et confirmation.

La couche physique offre deux types de services à travers les points d'accès au service PD-SAP « PHY Data – Service Access Point » et PLME-SAP « PHY Layer Management Entity – Service Access Point ». Le premier est utilisé par la couche MAC pour transmettre les trames via la couche physique ou pour recevoir les trames reçues par cette dernière. Le deuxième est utilisé pour le reste des services de gestion de la couche physique qui sont décrits dans le paragraphe 3.1.

La sous-couche MAC à son tour offre deux types de service à ses couches supérieures. Les services de transfert des données, à savoir ; l'envoi, la réception et la purge des paquets sont accessibles depuis les interfaces MCPS-SAP « MAC Common Part Sublayer – Service Access » par la sous-couche SSCS « Service-Specific Convergence Sub-layer ». Les services de gestion de la couche MAC sont accessibles par les couches supérieures via les interfaces MLME-SAP « MAC Layer Management Entity – Service Access ».

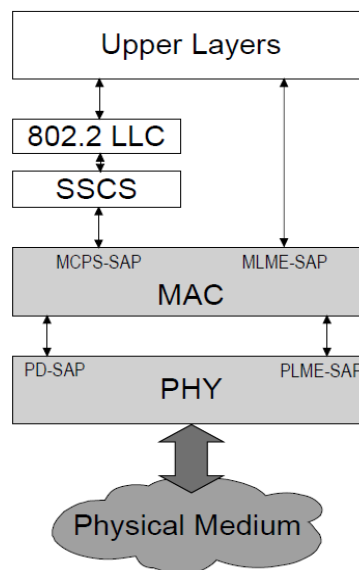


Figure 5 : Architecture d'un nœud LR-WPAN [5]

3.4. Rapport cyclique (Duty Cycle) dans le standard IEEE 802.15.4

La ressource d'énergie étant rare dans les RCSF, plusieurs protocoles MAC ont été proposés pour pallier ce problème en réduisant le rapport cyclique. Les protocoles S-MAC « Sensor-MAC » [65], T-MAC « Timeout-MAC » [66], U-MAC « Utilization MAC » [67] et OD-MAC « On-Demand MAC » [68] sont quelques exemples de ces protocoles MAC. Le standard IEEE 802.15.4 utilise aussi le mécanisme de Duty Cycle afin d'économiser de l'énergie et prolonger la durée de vie des nœuds du réseau. Cette option est disponible seulement dans le mode avec beacon activé lorsque les paramètres BO et SO sont différents ($SO < BO$). Cette alternance des deux modes de sommeil et de réveil est bien adaptée aux réseaux de capteurs sans fil, car dans la plupart des applications, les nœuds RCSF n'ont pas besoin de rester actifs tout le temps. Ils peuvent opérer pendant un temps très court pour transmettre ou recevoir les paquets. Ainsi, les nœuds peuvent profiter de ce temps d'inactivité en se mettant en mode veille pour conserver de l'énergie. Le coordinateur du PAN est le seul nœud qui doit rester actif tout le temps s'il gère plusieurs clusters, notamment dans le cas de la topologie d'arbre de clusters. Dans ce cas de figure, il doit être alimenté par une source d'énergie permanente.

Le rapport cyclique (Duty cycle « DC ») dans le cadre de la supertrame est calculé comme étant le pourcentage de la section active sur la durée totale de la supertrame. Cela se traduit par la division de la durée de la supertrame « SD » par l'intervalle du beacon « BI » selon l'équation suivante :

$$DC = \frac{SD}{BI} = 2^{SO-BO} \quad (IV)$$

Nous rappelons que selon le standard, une application typique pour les nœuds IEEE 802.15.4 généralement opère avec un rapport cyclique très faible (moins de 1 %). Mais le choix des valeurs des paramètres BO et SO est très important pour répondre aux besoins de l'application en terme de qualité de service. En effet, ces deux paramètres ont une influence sur la réactivité du réseau, et en conséquence, sur la qualité de service (voir le paragraphe 4.3).

3.5. Modèles analytiques pour le standard IEEE 802.15.4

Afin d'avoir une estimation théorique des performances du standard IEEE 802.15.4, un ensemble de modèles analytiques ont été proposés dans la littérature. Dans ce paragraphe, nous

allons présenter quelques exemples de ces derniers. Les auteurs dans [69], ont proposé un modèle analytique en utilisant la chaîne de Markov à temps discret et le modèle M/G/k queue. L'objectif de ce modèle analytique est d'évaluer l'impact de paramètres, tel que le taux de génération des paquets, le nombre de stations, la capacité de la mémoire tampon des stations, la taille des paquets et la taille de la période de sommeil, sur les performances du standard. D'autres modèles ont été proposés dans [70]–[72] pour étudier l'effet de la période de sommeil sur les performances du réseau en matière de consommation d'énergie et du débit. Un autre modèle analytique est proposé dans [73], permettant d'intégrer les paramètres de la qualité de service en termes de taux de délivrance des paquets « PDR 'packet delivery ratio' » et de latence des paquets, en plus de l'efficacité énergétique lorsque le mécanisme du « Duty Cycle » est activé.

4. Travaux de recherche sur la QoS au niveau de la couche MAC

Dans cette section, on s'intéresse aux protocoles MAC qui proposent des améliorations au standard IEEE 802.15.4, afin d'assurer une meilleur QoS. Ces protocoles se divisent en trois grandes familles : protocoles MAC avec contention, protocoles MAC sans contention et les protocoles MAC hybride. D'autres chercheurs préfèrent l'utilisation du concept de Cross-layer pour améliorer l'efficacité du standard.

4.1. Protocoles MAC avec contention

Les protocoles MAC avec contention utilisent la méthode d'accès CSMA-CA pour éviter les collisions. Cet algorithme se base essentiellement sur deux paramètres principaux, à savoir : l'exposant du backoff « BE » et la fenêtre de contention « CW ». Les protocoles de cette catégorie utilisent ces deux paramètres pour favoriser les trafics prioritaires et augmenter l'efficacité d'utilisation de la bande passante disponible.

Le mécanisme CoSenS [74] « Collect then Send burst Scheme » est implémenté au-dessus de la méthode d'accès CSMA-CA. L'idée de ce mécanisme est de séparer le temps de transmission en deux périodes. (i) La période d'attente, dans laquelle les routeurs collectent-ils les données de leurs nœuds fils et les stockent dans leurs files d'attente. La durée de cette opération dépend du trafic reçu. (ii) La période de transmission où les routeurs retransmettent en rafale les données stockées dans la première période. Le protocole offre un service différencié en adoptant deux politiques d'ordonnancement, à savoir, priorité fixe et l'algorithme EDF « Earliest Deadline First », pour organiser l'envoi des données stockées dans les files d'attente.

Les auteurs dans [75] proposent le mécanisme « Adaptive Backoff Exponent Mechanism » afin d'offrir un service différencié selon les priorités des paquets en question (haute priorité et faible priorité). Ce protocole se base essentiellement sur les paramètres « $macMinBE$ » et « $macMaxBE$ » et la fenêtre de contention « CW ». En effet, les paramètres qui seront utilisés dans la méthode d'accès CSMA-CA ne sont pas identiques pour les trafics de différentes priorités. Des paramètres spécifiques sont attribués à chaque classe de trafic. L'algorithme attribue un backoff inférieur pour le trafic prioritaire ($macMinBE_{HP} < macMinBE_{LP}$) afin d'améliorer la réactivité du réseau. En outre, le trafic prioritaire utilisera une fenêtre de contention « CW_{HP} » inférieure à celle du trafic non prioritaire « CW_{LP} » pour augmenter le temps d'écoute des trafics non prioritaires. En plus, les auteurs proposent l'usage d'un ordonnancement avec priorité stricte « Priority Queuing » pour réduire le délai d'attente dans les files d'attente.

Afin d'utiliser la bande passante de façon plus efficace, les auteurs dans [76] proposent un nouveau régime de transition d'état. En effet, ils estiment qu'avec la réduction de la valeur du paramètre « $macMinBE$ » pour certains nœuds, notamment ceux qui transmettent le plus souvent, le délai du backoff sera réduit pour ces derniers. Cela permet de limiter le gaspillage de bande passante dû aux intervalles de backoff utilisés par la méthode d'accès CSMA-CA pour éviter les collisions. Notant que la valeur du $macMinBE$ change dynamiquement en fonction des conditions de transmission.

Le mécanisme de différenciation par les fenêtres de contention CWD « Contention Window Differentiation », proposé dans [77], offre un service différencié multi-niveaux dans lequel les fenêtres de contention sont attribuées en fonction des priorités des nœuds. Ces dernières sont séparées en plusieurs classes de service. Ainsi, chaque trafic appartenant à une classe de service donnée, écoute le canal pendant un temps plus long que les trafics des classes plus prioritaires que lui.

Une méthode d'adaptation de trame FRT « Frame Tailoring » a été proposée dans [78]. Elle consiste en ce que les nœuds prioritaires utilisent un seul scan CCA⁴ « Clear Channel Assessment ». Pour cette raison, ladite méthode introduit un mécanisme pour éviter les collisions entre les paquets de données et des acquittements à cause de l'utilisation d'un CCA unique. Le temps d'attente de l'acquittement « t_{ACK} » est déterminé en fonction de la taille du paquet. Le

⁴CCA est une opération effectuée par l'algorithme CSMA-CA afin de vérifier si le canal est libre ou en cours d'utilisation. Le standard IEEE 802.15.4 utilise par défaut deux opérations CCA successives pour éviter des collisions potentielles avec les trames d'acquittement.

paramètre « term frame tail » est défini comme étant le reste de la division de la taille du paquet par la durée du backoff. Ainsi, une simple CCA ne pourra pas déclarer le canal libre pendant le temps d'attente de l'acquittement.

L'algorithme ABE « Adaptive Backoff Exponent » est proposé dans [79] après une étude qui a montré que la distribution inappropriée des exposants de backoff « BE » augmente la probabilité d'avoir des collisions. D'après les auteurs, ce comportement entraîne une dégradation des performances globales du réseau. Cet algorithme essaye de réduire la probabilité du choix des nombres de périodes de backoff identiques en se basant sur trois principes essentiels, à savoir : l'augmentation du rang de BE, la variation dynamique de la valeur de l'attribut « macMinBE » et la dépendance de la valeur de ce dernier de la contribution de chaque nœud dans le trafic global du réseau. Ainsi, les nœuds qui transmettent moins de données auront un accès plus rapide au médium.

4.2. Protocoles MAC sans contention

Le standard propose l'utilisation des GTS pour garantir un accès déterministe au médium sans fil. Cependant, ce mécanisme présente plusieurs faiblesses, entre autres :

- Les GTS peuvent s'épuiser rapidement, car le standard n'autorise que 7 GTS par supertrame au maximum. Ainsi, très peu de nœuds peuvent bénéficier d'un service garanti.
- L'allocation des GTS présente des limitations en termes d'efficacité et de déploiement avec un grand nombre de nœuds. Étant donné que la taille des GTS est définie pendant la phase de réservation en nombre de slots. L'utilisation partielle des GTS par leurs propriétaires qui transmettent des paquets à un taux très faible entraîne un gaspillage de la bande passante (p. ex. lorsque la fréquence d'envoi est très faible ou le temps nécessaire pour transmettre toutes les données temps-réel d'un nœud est inférieur à la durée du GTS).
- La réservation des GTS n'est pas garantie, car la trame de demande de GTS est envoyée par les nœuds dans la période avec contention CAP.

Plusieurs travaux de recherche ont été menés pour proposer des solutions à ces problèmes. Les auteurs dans [80] ont augmenté la taille de la supertrame tout en augmentant le nombre de GTS autorisé. L'objectif est de réduire le gaspillage de la bande passante et

d'améliorer le support de la qualité de service, notamment dans les réseaux où plusieurs nœuds ont des besoins de QoS.

Dans [81], les auteurs proposent la division de la taille des GTS en des slots de tailles plus petites que celle définie dans le standard afin d'utiliser la bande passante disponible de façon plus efficace.

Les auteurs dans [82] proposent une méthode d'allocation des GTS par priorité, de telle façon que les GTS les plus prioritaires seront réservés en premier. L'objectif de cette méthode est de résoudre le problème d'insuffisance des slots de GTS. Ainsi, les demandes de GTS sont classifiées en fonction de leurs priorités, ce qui permettra aux nœuds ayant des données critiques (temps-réel) d'avoir des GTS avant les autres nœuds avec des données moins prioritaires. Cette méthode permet de résoudre le problème de sous-utilisation des slot GTS alloués tout en réduisant le nombre des GTS disponibles.

L'algorithme GSA « GTS Scheduling Algorithm » [83] est un algorithme d'ordonnancement en ligne des GTS. Il distribue les GTS d'une transaction sur autant d'intervalles de beacon « BI » que possible en vue de satisfaire les exigences sur le plan du délai de la transaction. Cet algorithme permet la transmission des messages en rafale et des messages périodiques avec des contraintes temporelles même dans des réseaux surchargés.

Une méthode d'accès complètement déterministe est proposée dans [84] en supportant des slots temporels prédéfinis utilisés pour l'association des nœuds au coordinateur en temps-réel. Cette approche permet aussi d'éviter l'échec des demandes de GTS grâce à l'utilisation des PDS « Predefined Dedicated Timeslot ». De plus, les GTS sont distribués dans le PAN de façon à éviter les collisions dans les GTS entre les nœuds des différents clusters adjacents dans la topologie « cluster tree ». Le protocole proposé envisage aussi la réutilisation des GTS dans les clusters qui ne sont pas à portée radio les uns des autres (éloignés).

Les auteurs estiment dans [85] qu'une méthode d'allocation idéale des GTS doit être basée sur une bonne estimation du comportement d'utilisation des ces derniers par les nœuds. Ainsi, ils proposent le mécanisme AGA « Adaptive GTS Allocation Scheme » qui attribue des priorités de manière dynamique en se basant sur l'état d'utilisation des GTS, tout en prenant en considération l'équité et la garantie d'une faible latence. L'algorithme permet d'allouer des GTS aux nœuds qui en ont besoin et de réclamer les GTS alloués, mais non utilisés. Ce mécanisme est organisé en deux phases ; la phase de classification et la phase d'ordonnancement. Dans la

première phase, on attribue les priorités aux nœuds en fonction de leur utilisation des GTS. Dans la deuxième phase, un ordonnancement est fait de manière dynamique en allouant des GTS à des nœuds en fonction de leurs priorités. Ce mécanisme adopte l'algorithme AIMD [86] pour maintenir le numéro de priorité pour chaque nœud. L'ordonnancement quant à lui se base sur les numéros de priorités, la taille de la supertrame (c.-à-d. le paramètre BO) et la capacité des GTS dans la supertrame, à savoir, le nombre de GTS ne doit pas dépasser 7 et la taille de la période CAP doit être supérieure ou égale à « aMinCAPLength. ». Un modèle d'optimisation est proposé dans [87], qui calcule la priorité en se basant sur le taux de génération des paquets par les nœuds.

L'objectif principal de l'algorithme « knapsack » [88] est d'améliorer la manière avec laquelle les GTS sont réservés dans les protocoles basés sur le standard IEEE 802.15.4. Cet algorithme est destiné aux applications dans lesquelles un grand nombre de nœuds est déployé dans un réseau WiBaSe-Net « wireless body area sensor network ».

Une allocation implicite des GTS est proposée dans [89] suivant le mécanisme i-GAME « implicit GTS allocation mechanism ». Ce dernier offre la possibilité de partager le GTS par plusieurs nœuds selon la méthode « Round-Robin » en prenant en considération les spécifications du trafic et les besoins des flux en termes de délai. Les nœuds ayant besoin d'un service garanti envoient dans leurs demandes, à la fois, la spécification du trafic et les besoins de délai au coordinateur du PAN. Ensuite, ce dernier exécute un algorithme de contrôle d'admission en se basant sur les informations reçues et la quantité des ressources GTS disponibles.

4.3.Approches basées sur le choix des paramètres de la supertrame

Cette catégorie s'intéresse aux approches où la taille des périodes CAP et CFP est changé dynamiquement en fonction des conditions d'opération, d'où l'importance du choix des paramètres BO et SO. Des études ont montré l'impact de la variation de ces derniers sur les performances du standard IEEE 802.15.4 dans les deux périodes (CAP et CFP). Les auteurs dans [90] montrent que l'efficacité de l'algorithme CSMA-CA en terme de débit est améliorée en augmentant l'ordre de la supertrame « SO ».

Le mécanisme AGA [91] est un autre exemple de cette catégorie d'approches dites « Cross-Period ». Ce mécanisme permet d'ajuster la taille des périodes CAP et CFP selon la charge du trafic dans ces deux périodes. Un seuil est proposé pour les séparer. Dans le cas où le trafic dans la période CFP est faible, les ressources GTS sont transférées vers la période CAP pour filtrer les allocations des GTS non-nécessaires. Par contre, lorsque BO augmente,

l'algorithme suppose que la cause de cette variation est le nombre important de demandes de GTS envoyées, ce qui entraînera une définition d'un seuil plus stricte pour éviter le manque de GTS dédiés aux nœuds prioritaires.

Une étude de simulation réalisée dans [92] pour évaluer l'effet de ces deux paramètres sur le taux de perte des paquets et le débit de données utiles, montre qu'en fixant le paramètre SO, l'augmentation du paramètre BO diminue le débit de données effectives et augmente le taux de perte des paquets. Ceci vient du fait qu'en augmentant BO, le cycle rapport cyclique devient plus faible et, par conséquent, le temps disponible pour transmettre tous les paquets devient plus court.

Les auteurs dans [93] proposent un algorithme nommé PECAP « Priority-Based Delay Alleviation Algorithm », qui permet de configurer les deux paramètres BO et SO à la fin de la période CAP. En effet, les nœuds prioritaires demandent au coordinateur une extension de cette dernière pour réduire le temps de sommeil. Ainsi, cette partie additionnelle de la période CAP sera utilisée exclusivement par les nœuds prioritaires, ce qui limitera le nombre de contentions et donnera un accès au canal de transmission plus rapide à ces nœuds.

4.4. Cross Layer

Certains travaux de recherche ont opté pour le concept du Cross-layer afin améliorer les performances du standard IEEE 802.15.4. Le protocole TREN « Timely, Reliable, Energy-efficient and Dynamic » [94] est proposé pour les applications de contrôle industriel utilisant les RCSF. Les auteurs soutiennent l'approche basée sur les interactions complexes entre des protocoles de différentes couches du modèle OSI pour atteindre le maximum d'efficacité. Le protocole TREN permet la collaboration entre l'algorithme de routage, la couche MAC, et le contrôle de l'alimentation pour satisfaire la fiabilité et la latence désirées. Ceci est obtenu grâce à un processus d'optimisation avec un objectif de minimiser la consommation d'énergie. Dans ce protocole, le mécanisme de routage est divisé en routage statique pour manipuler communications inter-clusters, et le routage dynamique, pour gérer les communications inter-nœuds. Le protocole de routage est associé à une couche MAC hybride TDMA/CSMA-CA.

Dans [95], les auteurs proposent l'algorithme ADAPT « ADaptive Access Parameters Tuning » comme étant un Framework Cross-Layer distribué pour les RCSF qui intègre les standards IEEE 802.15.4 et ZigBee. L'objectif de cet algorithme est d'assurer un maximum de fiabilité avec une communication efficiente en énergie. ADAPT emploie un module d'adaptation

qui interagit directement avec toutes les couches de la pile ZigBee. Pour faciliter les interactions, ce module est réalisé sous la forme d'une composante verticale qui a un accès direct à chaque couche de la pile de protocoles. Les résultats de simulation ont montré que ce mécanisme donne de meilleures performances en termes de taux de délivrance de paquets et consommation d'énergie par message transmis.

Les approches basées sur le concept du Cross-layer permettent généralement d'obtenir de meilleures performances par rapport aux approches qui respectent la notion d'abstraction des couches définie dans le modèle OSI. Cependant, ces approches sont généralement plus complexes à mettre en œuvre et sont généralement moins compatibles avec les protocoles et normes existants et peuvent être, ainsi difficilement réutilisables. Cela donne naissance à un compromis entre performance et interopérabilité [96].

4.5. Le standard IEEE 802.15.4e

Le standard IEEE 802.15.4e [97] est un nouveau standard pour les réseaux WPAN. Il a été proposé comme un amendement de la couche MAC du standard original (IEEE 802.15.4-2011 [98]). L'objectif de cet amendement se résume dans l'amélioration de ce dernier et l'ajout de nouvelles fonctionnalités pour un meilleur support des applications industrielles et commerciales. Ce nouveau standard vise également à assurer la compatibilité avec le standard chinois CWPAN « Chinese Wireless Personal Area Network ». Selon la spécification du nouveau standard, il y a deux catégories d'amélioration de la sous-couche MAC. D'une part, une proposition des méthodes permettant le support de l'automatisation des processus industriels et des usines et, d'autre part, des améliorations fonctionnelles générales.

La couche physique PHY IEEE 802.15.4 est totalement préservée en vue de garantir une compatibilité avec les équipements existants. Aucun changement n'est requis dans le matériel, ce qui permet une réutilisation des appareils existants (p. ex. nœuds RCSF) avec une simple mise à niveau logicielle du protocole MAC utilisé.

En plus de la structure originale, ce standard propose de nouvelles structures de supertrame, plus particulièrement : la DSME multi-superframe et la supertrame basée sur le LL-Beacons.

4.5.1. Structure de la DSME multi-superframe

Le format de la structure multi-superframe DSME est défini par le coordinateur et annoncé par l'envoi d'un beacon amélioré EB « Enhanced Beacon » avec un descripteur du PAN DSME (IE « Information Element »). Cette multi-superframe est, en effet, un cycle de superframes répétées dans le temps. Chacune de ces superframes est constituée d'un Beacon, d'une période CAP et d'une période CFP (identique à la superframe originale).

La transmission du beacon amélioré « EB » et les échanges dans la période CAP utilisent un seul canal de fréquence commun, celui qui était utilisé dans l'association. Cependant, la diversité du canal (adaptation du canal ou saut du canal) est utilisée pour transmettre les données et les acquittements dans la période CFP. Le canal alloué pour le DSME-GTS, parmi les canaux disponibles dans le « ChannelPage », est utilisé pour la transmission des trames durant la période CFP.

4.5.2. Structure de la superframe basée sur le LL-Beacons

Cette superframe est constituée de trois types de slots. Le premier est réservé pour la trame LL-Beacon, qui est utilisée pour synchroniser les nœuds à la superframe. Le deuxième type est le slot de gestion. Ce dernier peut être d'un nombre de 0 ou 2 slots. Il est constitué d'un slot « downlink » et un autre « uplink ». Le dernier type est représenté par le reste des slots temporels. Chaque nœud est attribué exactement à un slot temporel. Le nombre de ces slots est défini dans l'attribut *macLLDNnum*. Si plusieurs nœuds sont attribués à un seul slot, ce dernier est appelé « slot temporel d'un groupe partagé », dans lequel on utilise un adressage simple de 8-bits.

5. Conclusion

Dans ce chapitre, nous avons présenté les différentes informations qui aident à la compréhension et la préparation de nos contributions. Ces dernières seront détaillées dans les chapitres qui suivent.

Nous avons montré que la fourniture de la QdS dans les RCSF n'est pas une tâche facile à cause des contraintes intrinsèques des nœuds RCSF et des problèmes liés à la nature de la communication sans fil. Nous avons centré notre étude sur le standard IEEE 802.15.4 qui fait l'objet de nos contributions. Les approches présentées dans l'état de l'art proposent des solutions pour améliorer la QdS en améliorant les différents mécanismes de ce standard, notamment la méthode CSMA-CA et le mécanisme des GTS. D'autres moyens ont été étudiés comme le choix des paramètres réseau et l'ordonnancement des données dans les files d'attente.

Malgré que les travaux de recherche existants apportent des améliorations significatives au standard IEEE 802.15.4, d'autres problèmes non résolus existent encore. En effet, la plupart des solutions proposées dans la littérature sont optimisées pour le modèle de communication Multi-source single-sink. Les résultats obtenus ne sont pas garantis pour un modèle de communication plus compliqué tel que le modèle Multi-source Multi-sink. Les failles que nous avons identifiées dans le standard augmentent le délai de communication de manière significative dans certains scénarios, notamment dans le cadre du modèle Multi-source Multi-sink, même lors de l'utilisation des slots garantis « GTS ». Ainsi, nous avons proposé un ensemble d'améliorations pour ce standard pour garantir un meilleur support de la QdS même dans un réseau à faible rapport cyclique. L'approche proposée est compatible avec la majorité des travaux présentés dans cet état de l'art, car leurs principes sont complémentaires.

La proposition protocolaire sera présentée dans le deuxième chapitre et évaluée dans le troisième chapitre par simulation et prototypage. La proposition d'un modèle de simulation ainsi que sa validation seront présentées dans le quatrième chapitre.

CHAPITRE II : Protocole FF-MAC, une Couche MAC pour une Meilleure QoS

Le présent chapitre est consacré à la description du protocole que nous avons conçu sous le nom FF-MAC « Fast-Forward MAC ». Un protocole que nous avons proposé en tant qu'amélioration du standard IEEE 802.15.4 et dont l'objectif est de fournir une meilleure qualité de service en termes de délai de communication pour les applications ayant des contraintes temporelles.

Ce chapitre sera organisé comme suit : nous allons situer dans un premier temps notre travail dans son contexte. Dans un second temps, nous allons décrire la problématique, puis nous aborderons les limitations identifiées dans le standard IEEE 802.15.4. Ensuite, nous présenterons notre protocole en détail : son fonctionnement, les solutions apportées par ce protocole aux problèmes identifiés ainsi que ses points forts permettant de résoudre le compromis énergie-délai. Un exemple d'application sera donné pour illustrer un cas d'utilisation du protocole proposé. Finalement, nous allons situer le protocole FF-MAC parmi les protocoles proposés dans la littérature.

1. Contexte de l'étude

1.1. Généralités

Dans ce travail, nous nous intéressons aux applications qui peuvent générer des trafics hétérogènes : un trafic avec contrainte temporelle, dite « temps-réel » et un trafic sans contrainte qui tolère un service de type « best effort ». La cohabitation de ces deux types de trafic nécessite la fourniture d'un service différencié. Dans la littérature, plusieurs protocoles MAC, notamment ceux de nature hybride, proposent des solutions à cette problématique. Notre travail s'inscrit dans cette catégorie. En effet, le protocole que nous avons conçu est une amélioration de la sous-couche MAC du standard IEEE 802.15.4. Ce standard, comme nous l'avons présenté dans le premier chapitre, présente une grande flexibilité se manifestant dans sa capacité de fournir des services différenciés selon les besoins des applications ou le besoin instantané des nœuds au cours du fonctionnement du réseau. Les services offerts par ce standard sont de deux types : un service non garanti dont l'accès au médium se fait avec contention et se gère par l'algorithme CSMA-CA (dans la période CAP) et un service déterministe de type TDMA dans lequel l'accès au canal est exclusif à certains nœuds pendant des tranches de temps appelées GTS (dans la période CFP).

Concernant le modèle de communication, une grande partie des protocoles proposés dans la littérature s'intéressent aux applications qui adoptent le modèle « Plusieurs sources vers une seule destination ». L'objectif dans ce genre de modèle est de recueillir les données collectées par des nœuds RCSF distribués dans une zone géographique vers un centre de traitement. Cependant il y a un autre modèle aussi important que le premier qui est le modèle « multi-source multi-sink ». Ce modèle sera présenté dans la section suivante. Le protocole FF-MAC apporte des améliorations qui touchent les deux modèles, mais surtout pour le modèle « multi-source multi-sink ».

1.2. Modèle Multi-source Multi-sink

Il s'agit d'un modèle de communication intéressant surtout dans le domaine des réseaux de capteurs et actuateurs sans fil (Wireless Sensor and Actuator Network « WSAN »). Il permet de rendre le réseau plus autonome et capable de prendre des décisions et faire des interventions par les nœuds RCSF d'une façon distribuée. En effet les nœuds RCSF, dans ce modèle, peuvent communiquer entre eux et échanger des informations sans passer par des stations de contrôle.

Malgré qu'il existe plusieurs travaux de recherche qui améliorent les performances des RCSF dans le modèle à une seule destination, les résultats obtenus peuvent ne pas s'appliquer à un modèle plus sophistiqué dans lequel, les capteurs peuvent avoir de multiples destinations potentielles. Ces nœuds destinataires peuvent être, à titre d'exemple, des actuateurs qui collectent des informations et prennent des décisions et des actions. Dans le contexte de notre étude, des contraintes temporelles peuvent être exigées pour ces échanges afin de garantir le bon fonctionnement et l'efficacité de ces applications.

2. Problématique et limitations identifiées dans le standard IEEE 802.15.4

2.1. Temps-réel

Le terme temps-réel dans un système peut être défini comme étant la possibilité qu'une de ces composantes répond à son environnement à temps. Mais la signification du terme « à temps » peut poser une ambiguïté qui peut la confondre avec le mot « rapidement ». Selon [99], une communication temps-réel pourra être définie tout simplement comme étant une communication suffisamment rapide dans le contexte où le système opère. Autrement dit, dans une communication temps-réel, le système doit être capable de répondre à une sollicitation donnée pour produire une réaction appropriée en un temps déterminé (déterminisme). Le délai de communication doit satisfaire les besoins définis par l'environnement, de tel sorte à ce que l'information, après acquisition et traitement, reste pertinente. Cette définition peut s'appliquer à tout système qu'il soit local ou distribué dans lequel les échanges entre ses composantes subissent des contraintes temporelles dont le respect est aussi important que l'exactitude du résultat.

Dans le contexte des réseaux de capteurs sans fil, notamment dans les applications industrielles, les nœuds RCSF ont été introduits comme un substituant aux capteurs connectés par des liaisons filaires. Malgré les avantages apportés par la communication sans fil, ses performances atteignent difficilement le niveau de qualité de service offert par la communication filaire. Ceci vient de la nature non fiable inhérente à ces technologies sans fil (p. ex. le taux de perte important, la gestion de l'accès au médium difficile, etc.).

La qualité de service temps-réel dans le domaine des réseaux représente la capacité de répondre aux exigences des applications en matière de délai de communication. Selon la tolérance aux contraintes exigées, on peut distinguer deux classes principales d'applications temps-réel : le temps-réel strict ou HRT « Hard Real-Time » et le temps-réel mou ou SRT « Soft Real-Time ».

Dans les deux classes, le système tient compte d'une borne du délai de bout-en-bout imposée. La différence entre les deux se manifeste dans le degré de tolérance au dépassement de cette borne. En effet, le système HRT doit supporter une borne déterministe du délai de bout en bout. Tout message qui arrive à sa destination après son échéance est considéré comme inutile et peut conduire à un échec du système. En conséquence, il peut amener à des situations critiques, voire catastrophiques. De l'autre côté, le système SRT, peut accepter une certaine tolérance au respect de cette échéance. Seulement une garantie probabiliste est requise.

Dans un réseau de capteurs sans fil, plusieurs types de trafics peuvent coexister, des trafics HRT, SRT et aussi les trafics qui tolèrent un service non garanti « Best effort ». Dans ce cas, le réseau doit être capable de gérer les ressources limitées, notamment en bande passante, de telle sorte qu'il garanti la qualité de service exigée par les trafics prioritaires, sans pour autant causer une famine de bande passante pour les trafics non prioritaires.

2.2. Compromis énergie-délai

Malgré l'importance du délai dans les systèmes temps-réel, les réseaux de capteurs sans fil ont plusieurs contraintes qui rendent le respect de la contrainte temporelle un véritable défi de recherche. Comme nous l'avons présenté dans le premier chapitre, l'efficacité énergétique est un facteur à ne pas négliger lors de la conception d'un protocole de communication pour les nœuds RCSF. Ceci vient de la nature des nœuds RCSF qui sont alimentés essentiellement par des piles⁵. Cela donne souvent naissance à un compromis entre ces deux considérations (c.-à-d. l'énergie et le délai). Selon le standard IEEE 802.15.4, le rapport cyclique (Duty Cycle) prévu pour les applications typiques des réseaux de capteurs sans fils est de moins de 1%. Cela veut dire que, à titre d'exemple, pendant 24 heures, l'activité du capteur ne doit pas dépasser les 14 minutes. Dans certaines applications, le capteur pourra se réveiller une fois par jour (voire moins fréquemment) pendant quelques millisecondes pour transmettre des données, puis il se remet en mode sommeil pour économiser de l'énergie. Ainsi, le rapport cyclique doit être soigneusement choisi en respectant les besoins de l'application ainsi que ses contraintes, notamment la contrainte temporelle, tout en réduisant le temps d'activité des nœuds RCSF au minimum possible afin maximiser leurs durée de vie.

Cependant, la réduction de l'activité des nœuds RCSF les rend moins réactifs avec leurs voisins. Ainsi, la question suivante se pose : comment permettre aux nœuds du réseau de

⁵ Des exceptions sont accordées pour quelques nœuds spéciaux qui peuvent être alimentés par des sources d'énergies permanentes (p. ex. le coordinateur du PAN).

communiquer et échanger leurs données dans des délais courts, tout en restant moins actifs pour économiser de l'énergie ? Plusieurs protocoles MAC ont été conçus pour résoudre cette problématique, dont une partie a été présentée dans le chapitre I.

Le standard IEEE 802.15.4 de son côté propose une solution à ce problème par l'utilisation des GTS. Néanmoins, le problème reste toujours présent. En effet, afin d'augmenter l'efficacité énergétique, les nœuds du réseau communiquent entre eux à travers leurs parents « le coordinateur ». Ce passage par le coordinateur est obligatoire, car le réveil simultané du nœud source et du destinataire final n'est pas garanti, voire interdite dans certains cas. Chaque nœud utilise le coordinateur comme un point de relai en lui envoyant ses données, puis ce nœud se met en sommeil. La réception par le destinataire se fait par le biais d'une demande initiée par ce dernier au point de relai, en l'occurrence le coordinateur. Ce processus entraînera évidemment un délai supplémentaire qui pourra être très important lorsque le rapport cyclique est très petit. Malgré que le standard propose une solution pour donner un accès rapide aux nœuds ayant des exigences spécifiques en bande passante ou ayant une contrainte temporelle, nous avons identifié un ensemble de failles importantes dans la solution proposée, que nous allons discuter dans la section suivante.

2.3. Limitations identifiées dans le standard IEEE 802.15.4

Le standard IEEE 802.15.4 propose une sous-couche MAC très flexible qui peut s'adapter à divers types d'applications. Pour les applications ayant des exigences en termes de délai et/ou de bande passante, le standard propose l'utilisation des GTS comme nous l'avons présenté plus en détail dans le premier chapitre. Il s'agit d'un accès au médium de type TDMA permettant une utilisation exclusive du canal, pendant un temps prédéfini, pour le nœud propriétaire. Cependant, ce mécanisme présente un ensemble de limites qui contribuent à la dégradation de la qualité de service et ne présente pas une solution satisfaisante au compromis énergie-délai.

Dans notre analyse, nous avons identifié ces problèmes comme étant relatifs à la structure actuelle de la supertrame. On peut les résumer dans les points suivants :

Premièrement, selon le standard IEEE 802.15.4, la période d'accès aléatoire avec contention (CAP) précède la période d'accès sans contention (CFP). Cela veut dire que les paquets à caractère non temps-réel (ne serait-ce que ceux qui gagneront l'accès au canal) seront transmis plus rapidement que les paquets qui subissent une contrainte temporelle. Le résultat de ce paradoxe s'aggrave quand la taille de la période active augmente, car la taille de la période CAP

augmentera en parallèle, et par conséquent, le temps d'attente sera plus important. Ce problème est illustré à la figure 6. La courbe montre la variation de l'écart entre l'envoi des paquets au début de la période CAP (slot 1) et celui dans le GTS (slot 15) en fonction du paramètre SO. Pour des raisons de simplification, nous supposons dans cet exemple que le Beacon occupe le slot 0, et que la période CFP contient un seul GTS composé d'un seul slot. Les valeurs sont calculées pour la couche Physique PHY 2.4 GHz. Nous pouvons constater que l'écart entre ces deux envois augmente de manière exponentielle lorsque la valeur du paramètre SO est incrémentée selon les équations (III) et (V) :

$$SD (\text{SuperframeDuration}) = (\text{aBaseSuperframeDuration} \times 2^{SO}) - (2 \times sd) \quad (V)$$

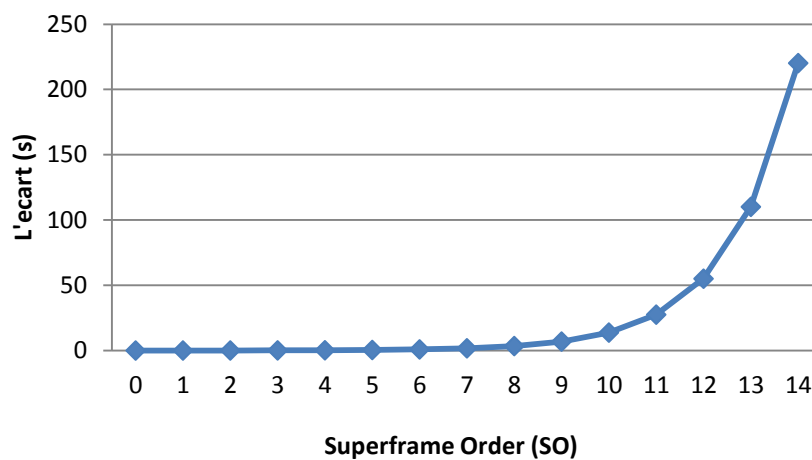


Figure 6 : Écart entre l'envoi du 1^{er} Paquet dans la CAP et le premier paquet dans la CFP

Pour résoudre ce problème, le standard propose une solution qui consiste à autoriser l'envoi des paquets dans la période CAP par les nœuds qui possèdent des GTS. Cependant, cette solution souffre de plusieurs limites : premièrement, l'envoi des données dans la période CAP n'est pas garanti, surtout dans le cas où la densité des nœuds dans le réseau est importante. Deuxièmement, cet envoi dans la CAP par les nœuds qui possèdent des GTS augmente le taux de contentions, ce qui introduit une dégradation de la performance de communication des autres nœuds. De plus, ceci entraînera une perte d'énergie supplémentaire. Finalement, lorsque les données sont envoyées avec succès dans la période CAP, le GTS réservé ne sera pas utilisé, ce qui représente un gaspillage de la bande passante. Donc nous pouvons conclure que cette solution n'est pas optimale.

Deuxièmement, un autre problème identifié est lié au modèle « multi-sources multi-sink » que nous avons présenté dans la section 1.2. Dans ce genre de scénarios, quand les nœuds sont

organisés dans une topologie en étoile, toutes les communications doivent passer à travers le coordinateur. Cette règle est très importante pour des fins d'efficacité énergétique. En effet en adoptant cette stratégie, les nœuds ne seront pas obligés de rester actifs pendant toute la période d'activité. Mais, au moment où un nœud souhaite envoyer des paquets, il active son module radio dans cette période, puis il tente d'accéder au canal pour envoyer ses données au coordinateur, et après le succès de la transmission, il retourne à l'état de sommeil jusqu'à la prochaine supertrame. Pour le destinataire, il attend la réception du Beacon pour savoir s'il a des données en attente chez le coordinateur. Dans ce cas, si son adresse est annoncée dans le Beacon de la supertrame suivante, il essaiera de récupérer les données par l'envoi de la commande « Data request ». Sinon il se mettra en sommeil pour conserver de l'énergie. Ce mécanisme est aussi appliqué aux envois dans la période CFP, car seul le nœud propriétaire du GTS et son coordinateur, qui doivent rester actifs pendant toute la durée de ce GTS.

Trois remarques peuvent être tirées de ce mécanisme :

- Le stockage des données dans le coordinateur pendant toute la supertrame (même pour les paquets envoyés dans les GTS). Cela signifie que le délai de bout-en bout sera relatif à la taille de la supertrame. Une taille qui pourra être longue dans des applications typiques de RCSF à cause de la longue période de sommeil nécessaire pour conserver de l'énergie⁶.
- La réception se fait par la méthode d'accès CSMA-CA. Ce mécanisme aléatoire ne présente aucune garantie que les données seront reçues dans la supertrame suivante ce qui peut introduire un délai supplémentaire.
- Le coordinateur ne peut annoncer dans son beacon que 7 adresses en attente dans sa file d'attente. Ainsi, les trames prioritaires peuvent être cachées par les trame non prioritaire⁷ étant données qu'il n'y a aucun mécanisme pour dans le standard permettant de gérer cette liste en fonction des priorités.

⁶ Un rapport cyclique de moins de 1 % selon le standard IEEE 802.15.4.

⁷ Dans le standard IEEE 802.15.4, les trames transmises dans la période CAP sont stockées dans la file d'attente avant les trames prioritaire transmises dans les GTS. La gestion FIFO de la file d'attente favorisera les trames CAP par rapports aux trames CFP.

3. Amélioration du standard IEEE 802.15.4 pour une meilleure qualité de service temps réel.

3.1. Améliorations proposées

Afin de proposer des solutions aux problèmes exposés auparavant, nous avons pensé à une restructuration de la supertrame tout en essayant de rester dans la même logique du standard IEEE 802.15.4. Les améliorations se résument dans les points suivants :

- Changement de l'ordre des périodes CAP et CFP au sein de la section d'activité : nous avons choisi de mettre la période CFP au début de la supertrame.
- Introduction de la période D-CFP (CFP Dynamique) entre la CFP et la CAP : elle est utilisée pour garantir la réception des trames prioritaires
- Ajout d'une nouvelle trame MAC de gestion : cette trame est envoyée par le coordinateur pour décrire la forme de la période D-CFP.
- Allocation plus efficace de la taille des D-GTS (GTS alloué Dynamiquement) : la durée du D-GTS correspond à l'espace nécessaire pour l'envoi des données en attente.
- Création d'une file d'attente virtuelle⁸ pour les trames temps-réel : elle permet de protéger ces derniers du manque d'espace dans la file d'attente principale.

3.1.1. Changement de l'ordre des périodes CAP et CFP

Afin de résoudre les problèmes identifiés dans le format de la supertrame actuelle de standard IEEE 802.15.4, nous avons procédé dans un premier temps à un changement de l'ordre des deux périodes CAP et CFP à l'intérieur de la période active [100]. Nous avons placé la période sans contention « CFP » au début de la supertrame juste après la réception du Beacon. Quant à la période avec contention, nous l'avons placée après la période CFP, en occupant tout le reste de la période active de la supertrame. La période d'inactivité (Sleep), quant à elle, reste à sa place localisée à la fin de la supertrame. Cette nouvelle structure, illustrée dans la figure 7, propose deux améliorations essentielles. La première étant liée à la rapidité d'envoi des données temps-réel. En effet, ces dernières sont délivrées au début de la supertrame sans attendre la fin de la CAP. Le nœud source n'a plus besoin de stocker ses données temps-réel dans sa mémoire et de différer leur transmission au GTS localisé à la fin de la supertrame. La deuxième amélioration se

⁸ Cette idée a été proposée à l'issue des résultats expérimentaux et sera discuté dans le chapitre III.

manifeste dans la possibilité d'achever la transmission au début de la supertrame et de libérer la mémoire du nœud RCSF plus rapidement.

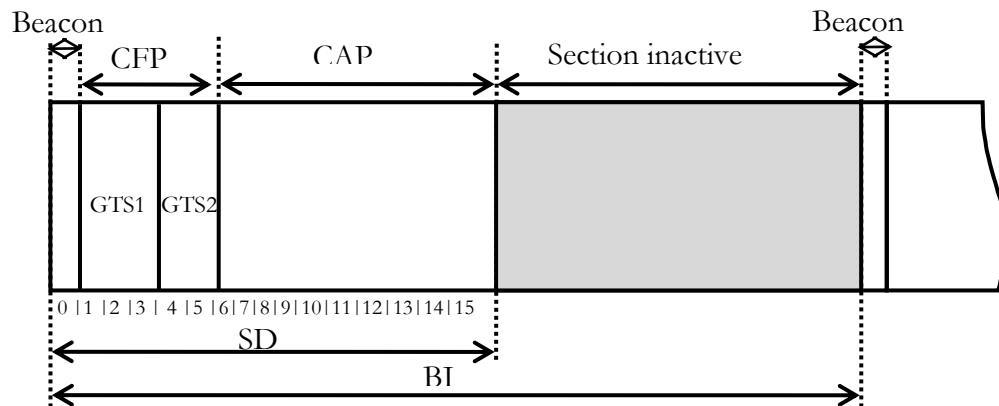


Figure 7 : Nouvel ordre des deux périodes CAP et CFP dans la nouvelle supertrame

3.1.2. Ajout de la nouvelle période D-CFP

Le nouveau positionnement des périodes CFP et CAP nous offre des avantages considérables. Dans le modèle multi-source multi-sink, où les nœuds communiquent entre eux via leur coordinateur comme un point de relai, les règles définies dans le standard IEEE 802.15.4 imposent que l'envoi et la réception des données soit dans des supertrames différentes. Ceci s'applique surtout pour les données transmises dans la période CFP puisqu'elle est placée à la fin de la période active. La structure améliorée de la supertrame nous permet de résoudre ce problème, car selon cette nouvelle forme, l'envoi des données temps réel dans la CFP est suivi d'une période CAP. Ceci nous donne la possibilité de retransmettre les données à leurs destinations finales dans la période CAP de la même supertrame. Nous avons proposé dans [100] et [101] deux approches pour profiter de cette nouvelle possibilité.

La première consiste à essayer de retransmettre dans la période CAP les données reçues par le coordinateur durant les GTSs. En effet, après la fin de la période CFP, le coordinateur envoie une trame MAC spécifique qui contient une liste des adresses destinations des nouveaux paquets qu'il a reçus dans la période CFP. Les nœuds ayant des données en attente (soit ceux annoncés dans le Beacon ou ceux annoncés dans la commande MAC) essayeront de les récupérer dans la période CAP par l'envoi de la commande MAC « Data Request ». Cette méthode permet de donner la possibilité de recevoir les données prioritaires dans la même supertrame. Néanmoins, aucune garantie n'est assurée, car l'accès au médium se fait avec contention selon la

version slottée de l'algorithme CSMA-CA. Pourtant, une amélioration considérable du délai moyen est constatée dans les résultats obtenus (présentés dans le troisième chapitre).

La deuxième approche que nous avons proposée est une amélioration de la première. L'objectif de cette amélioration est de supprimer l'aléa de la méthode d'accès CSMA-CA et le remplacer par un accès purement déterministe. En effet, la clé de cette solution consiste à générer de manière dynamique une nouvelle période que nous avons nommé D-CFP « Dynamic Contention Free Period ». Cette période est créée automatiquement par le coordinateur lorsqu'il reçoit des données dans la période CFP. En fonction des paquets envoyés dans la CFP, le coordinateur crée un ordonnancement d'un ou plusieurs D-GTS « Dynamically allocated GTS » pour former la D-CFP. Ces D-GTS sont tous de type « receive only », car l'objectif de cette période est de retransmettre les données temps-réel, reçues par le coordinateur dans la CFP, à leurs destinations respectives.

Cette deuxième solution, que nous avons adoptée pour le protocole FF-MAC, fournit un envoi en temps-réel avec des délais stables et bornés. Ces conclusions sont extraites de la logique du fonctionnement des deux approches ainsi que des résultats expérimentaux réalisés qui seront discutés dans le chapitre suivant.

3.1.3. Trame PRTPA.

Il s'agit d'une nouvelle trame MAC dédiée à la notification des informations relatives à la période D-CFP. Cette trame est destinée aux nœuds associés au coordinateur. Elle contient les adresses des nœuds concernés, la taille des D-GTS ainsi que le séquençement de ces derniers. Le format de cette trame est détaillé dans la section 3.1.3.

3.2. Généralités sur le protocole FF-MAC :

FF-MAC signifie « Fast-Forward MAC », est une couche MAC à retransmission rapide. Cette description vient du fait que notre protocole, qui est une amélioration de la couche MAC du standard IEEE 802.15.4, a comme objectif d'accélérer la transmission des paquets ayant des contraintes temporelles. Ce protocole permet d'envoyer et de recevoir ces dernières dans la même supertrame grâce à la version améliorée de la supertrame.

Afin de permettre un accès rapide au média partagé pour les nœuds ayant des contraintes en termes de délai de communication ou de bande passante, ce protocole utilise un ordonnancement déterministe en adoptant la deuxième forme améliorée de la supertrame illustrée dans la figure 8. La première partie de cet ordonnancement est représentée par la période CFP,

qui n'est autre que la période CFP définie par le standard, mais localisée au début de la supertrame. Cette période pourra être utilisée pour l'envoi et/ou la réception des données par les nœuds qui ont réussi à réserver des GTS. La deuxième partie, si elle existe, est la période dynamique D-CFP, où les nœuds ayant des D-GTS reçoivent des données temps-réel précédemment reçues par le coordinateur dans la période CFP. Ces deux périodes sont séparées par l'envoi d'une trame MAC spéciale, appelée PRTPA « Pending Real-Time Packets Advertisement », annonçant l'organisation et les dimensions de la D-CFP. Les informations contenues dans cette trame sont : les adresses courtes des nœuds ayant des données temps-réel en attente, la taille de chaque D-GTS selon la quantité de données dans la file d'attente du coordinateur et l'ordre des D-GTS. Les D-GTS ne peuvent être utilisées que dans le sens descendant (c.-à-d. du coordinateur vers les destinations annoncées) et pour transmettre les acquittements dans le sens inverse, le cas échéant.

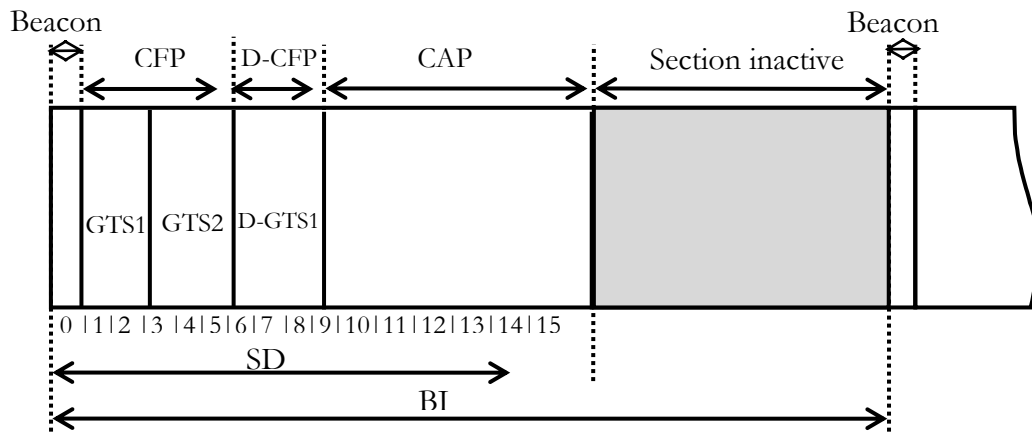


Figure 8 : Exemple de la nouvelle structure de la supertrame selon le protocole FF-MAC

La période « D-CFP » disparaît automatiquement au moment où il n'y a plus de données temps-réel dans le coordinateur, ou quand il n'y a pas de GTS de transmission (transmit only) dans la CFP. Lorsqu'il n'y a plus de CFP, le protocole FF-MAC devient totalement compatible avec le standard, car la trame PRTPA disparaît aussi et il n'en reste que la période CAP.

La période CAP, dont une partie peut se transformer en D-CFP, est une période à accès aléatoire gérée par une version particulière du fameux algorithme CSMA-CA. Cette période est utilisée par les nœuds n'ayant aucun besoin spécifique, ceux qui n'ont pas réussi à réserver des GTS ou pour l'envoi des commandes MAC.

Les échanges au cours de la période active de la supertrame sont illustrés globalement dans le diagramme de séquence de la figure 9.

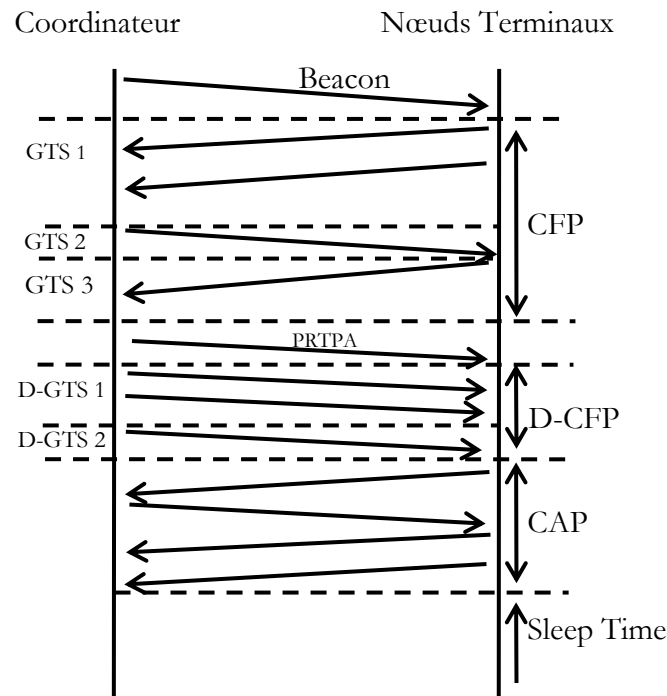


Figure 9 : Diagramme de séquence décrivant le cadencement des échanges pendant la section active de la supertrame

3.3.Format de la trame PRTPA

La trame PRTPA est un cinquième type de trame MAC (outre que DATA, ACK, BEACON et COMMANDE). Ainsi, pour la distinguer des autres trames, nous avons choisi la première valeur disponible parmi celles réservées, la valeur binaire « 110 » (6 en décimale), codée sur 3 bits dans le sous-champ « type de trame » du champ « contrôle de la trame ». En effet, dans le standard IEEE 802.15.4, la première valeur disponible est la valeur binaire « 100 ». Cependant, dans la nouvelle version IEEE 802.15.4^e [97], les valeurs « 100 » et « 101 » ont été allouées aux nouvelles trames « LLDN » et « Multipurpose » respectivement. Les différentes valeurs allouées dans le standard en plus de la valeur que nous avons réservée pour le paquet PRTPA sont indiquées dans le Tableau 3.

Frame Type value $b_2b_1b_0$	Description
000	Beacon
001	Data
010	Acknowledgment
011	MAC command
100	LLDN (IEEE 802.15.4 ^e)
101	Multipurpose (IEEE 802.15.4 ^e)
110	PRTPA (FF-MAC)
111	Reserved

Tableau 3 : Valeurs du champ « Type de trame »

La trame PRTPA adopte la forme générale des trames MAC définies dans le standard IEEE 802.15.4. Elle est constituée de trois sections majeures ; l'en-tête MAC, la charge MAC et le pied de la trame MAC. Plus de détail est explicité dans la figure 10.

Octets : 2	1	4	0/5/6/10/14	variable	2
Contrôle de la trame	Numéro de Séquence	Champs d'adressage	En-tête auxiliaire de Sécurité	Charge de la trame PRTPA	FCS
MHR				MAC Payload	MFR

Figure 10 : Format de la trame PRTPA

3.3.1. En-tête MAC :

L'en-tête MAC est formé par un ensemble de champs qui contiennent les informations nécessaire pour traiter correctement la trame dans la couche MAC du destinataire.

Le champ de contrôle de trame « Frame Control » occupe deux octets. Il contient les informations qui permettent définir le type de la trame, le type d'adressage utilisé, la sécurité, etc. Les valeurs choisies pour la trame PRTPA sont représentées dans la figure 11. Les 3 premiers bits de ce champ doivent contenir la valeur que l'on a choisie pour elle (c.-à-d. $b_2b_1b_0 = 110$ en binaire). Le bit b_3 , indique si la sécurité est activée ou non, et par conséquent, il indique si l'en-tête auxiliaire de sécurité est présent dans l'en-tête MAC ou non. Les bits b_4 (Frame Pending) et b_5 (Ack request) doivent être mis à zéro et ignorés à la réception. Pour le bit b_6 « PAN ID Compression », il doit avoir une valeur de 1, car les nœuds auxquels cette trame est destinée doivent être dans le même PAN, d'où la possibilité de compresser l'identifiant du PAN « PAN ID ». Quant aux bits réservés b_7 , b_8 et b_9 , nous allons les utiliser pour annoncer le nombre de D-GTS que le coordinateur a réservé pour la D-CFP et dont les descripteurs sont inclus dans la charge de cette trame. La valeur zéro indiquera que ceci est une trame vide et déclenchera le début immédiat de la période CAP. Les bits b_{12} et b_{13} « Frame Version » ne doivent être positionnés à la valeur « 01 » que si la sécurité est activée. Le mode d'adressage source doit indiquer une adresse courte de 16-bits « Short Address » ($b_{15}b_{14} = 10$) et le mode d'adressage destination doit être égal à 0 ($b_{11}b_{10} = 00$), car l'adresse destination n'est pas disponible dans l'en-tête de la trame (trame destinée à tous les nœuds).

Bits : 0-2	3	4	5	6	7-9	10-11	12-13	14-15
110	0/1	0	0	1	D-GTS count	00	Frame Version	10

Figure 11 : Valeurs des bits du champ de contrôle de la trame PRTPA

En ce qui concerne le champ de numéro de séquence, il doit contenir la valeur actuelle de l'attribut $macBSN - 1$. En effet, le numéro de séquence de la trame PRTPA doit être identique à celui du dernier Beacon diffusé pour être capable de vérifier si cette trame appartient à la supertrame actuelle. Cette information pourra aussi être utile pour détecter rapidement la perte du Beacon et, éventuellement, continuer à transmettre dans le reste de la supertrame⁹ (c.-à-d. D-GTS ou CAP).

Le champ adressage est réduit au minimum, il ne contient que l'adresse source courte de 16-bits et l'identifiant source du PAN. Les informations d'adressage de destination sont absentes, car l'information sur le type de trame est suffisante pour que la couche MAC des nœuds du réseau réalise qu'il s'agit d'une trame diffusée.

3.3.2. Charge de la trame PRTPA

L'information utile transportée dans la trame PRTPA correspond à l'ordonnancement de la période D-CFP. Elle donne le détail sur les adresses courtes des nœuds propriétaires des D-GTS, les tailles de ces derniers et la façon dont ils sont ordonnés. La taille de cette charge dépend de la valeur du « D-GTS count » inclus dans le champ contrôle de trame (bits 7 à 9). Lorsque la valeur du « D-GTS count » est différente de zéro, chaque descripteur de D-GTS « D-GTS Descriptor » prend la forme illustrée dans la figure 12. Sa taille est un multiple de 3 octets (2 octets pour l'adresse courte et 1 octet pour la taille du D-GTS) :

$$\text{Taille de la charge PRTPA} = \text{« D-GTS count »} \times 3 \text{ Octets} \quad (\text{VI})$$

Bits : 0-15	16-23
L'adresse courte du nœud	La taille du D-GTS (en nombre de backoff)

Figure 12 : Format d'un descripteur D-GTS

3.3.3. Pied de la trame

Le champ FCS est un champ de deux octets. Il contient le résultat de calcul du CRC (contrôle de redondance cyclique) suivant la norme (16-bit ITU-T). Son rôle se limite à vérifier de l'intégrité de la trame afin de détecter les erreurs de transmission.

⁹ À condition que le changement des paramètres de la supertrame pendant le fonctionnement du réseau ne soit pas autorisé.

3.4. Fonctionnement détaillé du protocole FF-MAC

Dans le protocole FF-MAC, la supertrame commence par la diffusion d'une trame Beacon identique à celle définie dans la spécification du standard IEEE 802.15.4. Il contient tous les paramètres qui identifient les dimensions de la supertrame ainsi que d'autres informations supplémentaires (les adresses des nœuds ayant des données en attente, information sur les GTS, etc.). La seule différence se manifeste dans le sous-champ « Final CAP Slot » qui indique, selon le standard, le dernier slot de la période CAP. Il est localisé dans le champ de spécification de la supertrame dans la trame Beacon (voir figure 13). Dans un cas général, ce paramètre est égal à 15 quand la CAP occupe toute la section active de la supertrame. Mais la valeur de ce paramètre change dès l'apparition des GTS. Le cas échéant, le champ « Final CAP Slot » précise la limite de la période CAP. Cependant, ceci n'est plus le cas dans notre protocole, car la fin de la CAP correspond toujours à la fin de la section active de la supertrame. Autrement dit, la valeur de ce paramètre sera toujours égal à 16, car cette période est placée à la fin de la période active. Par conséquent, nous n'aurons plus besoin d'annoncer une valeur constante et prédéfinie dans le Beacon. Pourtant, le début de la CAP peut changer lors de l'apparition de la période CFP, c'est la raison pour laquelle nous allons réutiliser ce champ pour identifier le début de la CAP au lieu de sa fin. Dans notre cas, nous l'avons renommé « First CAP Slot ». Sa position dans le champ de spécification de la supertrame dans l'en-tête MAC du Beacon est indiquée dans la figure 13. S'il n'y a pas de CFP, ce paramètre sera positionné à 15 pour indiquer que la période CAP occupe la totalité de la période active. Le choix de cette valeur est justifié par ; premièrement, cette valeur est interdite lors de la présence de la période CFP (la raison est expliquée dans la section 3.5). Deuxièmement, cette valeur permet à ce protocole de rester compatible avec le standard dans les réseaux qui n'autorise pas les GTS (GTS peut avoir une valeur « FALSE »). L'objectif est de pouvoir utiliser la même implémentation de notre protocole dans les réseaux qui utilise le standard IEEE 802.15.4 sans support du mécanisme des GTS.

Bits: 0–3	4–7	8–11	12	13	14	15
Beacon Order (BO)	Superframe Order (SO)	Final CAP Slot ➔ 'First CAP Slot' (FF-MAC)	Battery Life Extension (BLE)	Reserved	PAN Coordinator	Association Permit

Figure 13 : Format du champ de spécification de la supertrame défini dans le standard IEEE 802.15.4 et la modification apportée par le protocole FF-MAC

La période CFP commence directement après la réception du Beacon, cette période est utilisée exclusivement par les nœuds qui ont réussi à réserver des GTS. Les échanges des paquets se font selon les directions définies au moment de la réservation. Toutes les communications

dans cette période sont effectuées seulement avec le coordinateur, car le reste des nœuds doit être en mode de sommeil pour préserver la ressource d'énergie. Si les paquets reçus par le coordinateur sont destinés à un autre nœud du réseau, ils seront stockés dans sa file d'attente et à la fin de la période CFP, ils seront annoncés dans la trame PRTPA. Les échanges se font de manière directe dans les GTS sans utiliser l'algorithme CSMA-CA.

À la fin de la période CFP, le coordinateur effectue un bilan des paquets reçus dans cette période et crée un ordonnancement à la volée pour pouvoir transférer les paquets ayant des contraintes temporelles avant le commencement de la période CAP. Cette communication sera effectuée dans une période dynamique nommée « D-CFP ». L'ordonnancement fournit les tailles exactes des D-GTS nécessaires pour délivrer tous les paquets à caractère « critique » à leurs destinations respectives sans gaspiller la bande passante. Le calcul tient compte du nombre de paquets pour chaque destination (N_i), les espacements inter-frames (IFS) qui sont obligatoires afin de laisser suffisamment de temps à la couche MAC du nœud destinataire pour traiter les paquets remontés par sa couche physique (PHY). Le nombre d'espacements inter-frames nécessaires pour chaque destination est de $N_i - 1$, car après la réception du dernier paquet, le nœud aura suffisamment de temps pour le traiter en dehors du D-GTS. Les temps d'attente et de transmission des acquittements doivent être aussi pris en considération quand la transmission de ce dernier est requise. Ainsi, pour chaque D-GTS, la taille pourra être calculée de la manière suivante (en supposant que les paquets ont une taille identique pour simplifier) :

$$\text{Taille du D-GTS}_i = \left[N_i \times Q_{\text{données}} \frac{SR}{DR} \right] + [(N_i - 1)] \times IFS + a \times N_i \times [t_{\text{Ack}} + Q_{\text{Ack}} \frac{SR}{DR}] \quad (\text{VII})$$

Tel que :

N_i est le nombre de paquets en attente pour un nœud i . Q_x est la taille du paquet de type « x » en nombre de bits (y compris l'en-tête de la couche physique). SR et DR sont le débit de symbole « Symbol Rate » et le débit binaire « Data Rate » respectivement. Le rapport $\frac{SR}{DR}$ est introduit dans l'équation afin de convertir le nombre de bits, notamment dans la taille des données, en nombre de symboles. Le résultat de ce rapport varie selon la couche PHY et la technique de modulation adoptée. À titre d'exemple, pour la couche PHY 2.4 GHz, ce rapport est égal à :

$$\frac{SR}{DR} = \frac{62,5 \text{ kSymb/s}}{250 \text{ kbit/s}} = \frac{1}{4} \text{ Symbol/bit} \quad (\text{VIII})$$

Chaque symbole correspond à 4 bits dans la couche physique PHY 2.4 GHz.

IFS est une variable qui contient la valeur de l'espacement inter-frame. Seulement deux valeurs sont possibles en fonction de la taille de la trame transmise. SIFS, d'une durée minimale de

$macMinSIFSPeriod$ symboles, pour les trames d'une taille de MPDU jusqu'à $aMaxSIFSFrameSize$ octets et LIFS, d'une durée minimale de $macMinLIFSPeriod$ symboles, pour les trames d'une taille qui dépasse les $aMaxSIFSFrameSize$ octets. Les valeurs des constantes utilisées sont regroupées dans le Tableau 4.

$aMaxSIFSFrameSize$	$macMinSIFSPeriod^{10}$	$macMinLIFSPeriod^1$
18 octets	12 symboles	40 symboles

Tableau 4 : Valeurs de quelques constantes MAC pour le calcul des IFS

Le paramètre « a » indique si la transmission d'un acquittement est requise ou non. Il prend les valeurs 0 (Ack non requis) ou 1 (Ack requis). Le dernier paramètre est t_{Ack} qui détermine le nombre de symboles que le nœud doit attendre pour la réception d'un acquittement (voir l'inégalité (IX) pour les bornes inférieures et supérieures). Nous supposons dans ce calcul le cas le plus défavorable où le nœud devra attendre une durée de 32 symboles avant de recevoir cet acquittement (c.-à-d. $aTurnaroundTime + aUnitBackoffPeriod$).

$$aTurnaroundTime \leq t_{ack} \leq aTurnaroundTime + aUnitBackoffPeriod \quad (IX)$$

Remarque : Le résultat du calcul est donné en nombre de périodes de backoffs, car dans le mode « Beacon-enabled », les transmissions doivent être alignées sur les frontières des backoffs.

Après avoir effectué ce calcul, le coordinateur diffuse cet ordonnancement dans la trame PRTPA dont le format a été détaillé dans la section 3.3. Simultanément, les nœuds du réseau doivent se réveiller et remettre leur émetteur-récepteur en mode réception « RX_ON » pour pouvoir capter la trame PRTPA. Cette trame transmise par le coordinateur permettra aux nœuds du réseau d'avoir des informations sur la structure de la période D-CFP.

Afin de garantir la réception de la trame PRTPA, les récepteurs des nœuds fils doivent rester activés pour une durée maximale de $(aTurnaroundTime + aUnitBackoffPeriod)$ ou jusqu'à la réception de cette trame. Chaque nœud qui reçoit la trame PRTPA doit suivre la démarche illustrée dans le diagramme de la figure 14.

¹⁰ Le nombre de symboles est identique pour toutes les couches physiques. Cependant la durée des IFS dépend de la couche physique utilisée, car ces derniers n'offrent pas tous les mêmes débits de symbole.

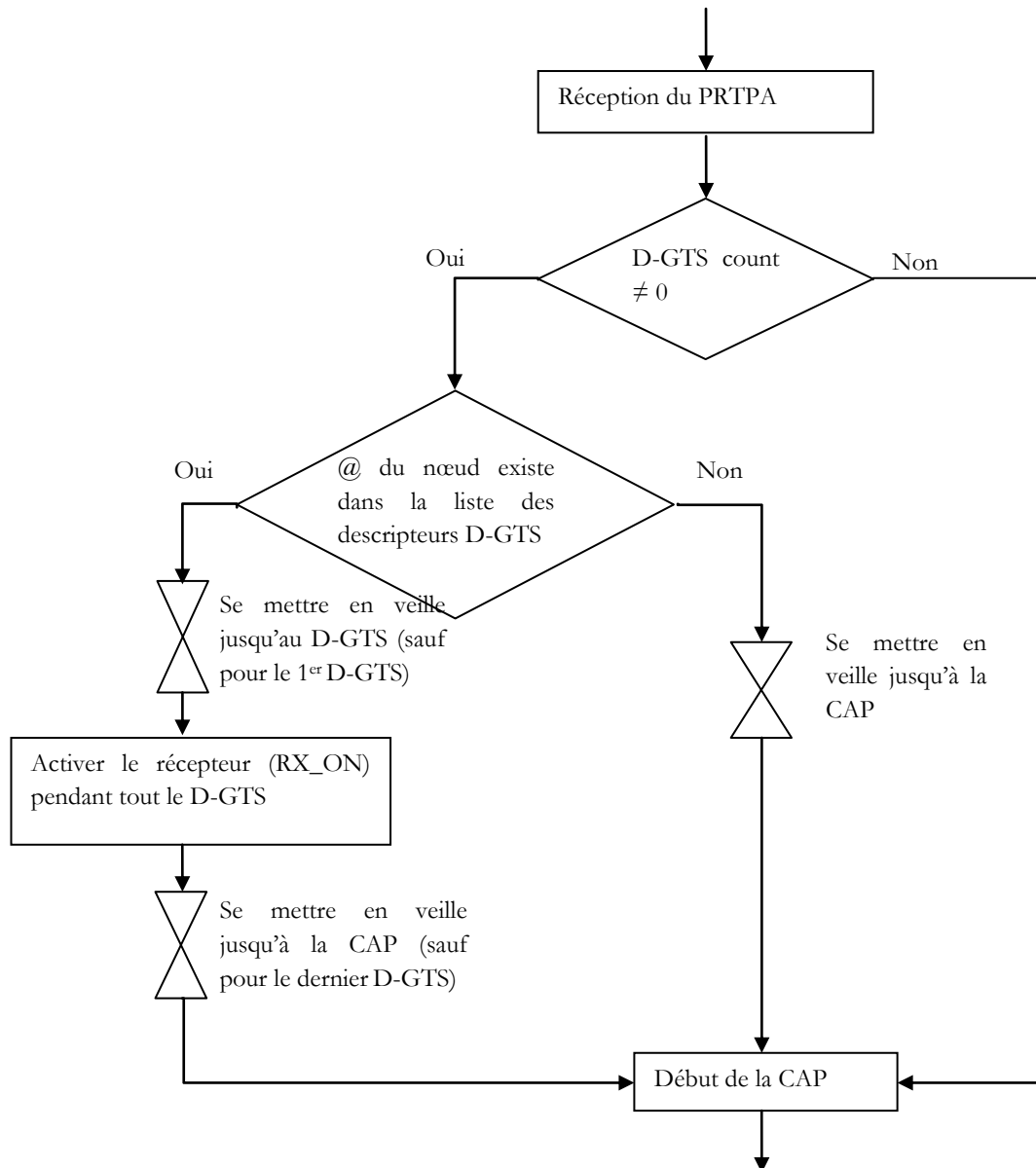


Figure 14 : Algorithme suivi par un nœud terminal lors de la réception de la trame PRTPA

Si la valeur du champ « D-GTS count » est nulle, la période CAP doit commencer immédiatement et suivre la même procédure décrite dans la spécification du standard IEEE 802.15.4. Si son adresse est annoncée dans le Beacon, le nœud en question envoie tout d'abord une commande MAC pour demander ces données au coordinateur. Ensuite, ce nœud essaye d'envoyer ses données s'il en a. Tout accès au canal dans cette période se fait suivant la version slottée de l'algorithme CSMA-CA.

Si la valeur du « GTS count » est différente de zéro, le nœud parcourt la liste des descripteurs de D-GTS en vue de chercher s'il y a un D-GTS pour lui ou non. Selon le résultat de la recherche, deux scénarios sont envisagés : le premier scénario est celui dans lequel le nœud ne

trouve pas son adresse dans la liste. Dans ce cas de figure, le nœud calcule la taille de la D-CFP puis se met en veille jusqu'au commencement réel de la période CAP. Dans le deuxième scénario où le nœud trouve son adresse dans la liste, il doit se mettre en veille jusqu'à son D-GTS, puis il met son émetteur-récepteur en mode réception pour recevoir les trames en provenance du coordinateur. À la fin du D-GTS le capteur se met en veille jusqu'au début de la période CAP s'il a des données à envoyer, ou jusqu'au début de la supertrame suivante dans le cas contraire.

La communication dans les D-GTS est unidirectionnelle, du coordinateur vers les nœuds. Mais il peut y avoir un envoi des acquittements dans le sens inverse si nécessaire. L'envoi des paquets se fait directement sans passer par la méthode d'accès CSMA-CA, car l'utilisation de chaque slot est exclusive au nœud désigné par le coordinateur. Chaque nœud qui dispose d'un D-GTS doit se mettre en mode réception pendant toute la durée de son D-GTS afin de recevoir tous les paquets envoyés par le coordinateur.

3.5. Protection de la période CAP :

Malgré que les périodes CFP et D-CFP ont une priorité par rapport à la période CAP, notre protocole doit protéger cette dernière afin de ne pas causer une famine de bande passant pour les nœuds non prioritaires. Dans notre protocole FF-MAC, les périodes CFP et D-CFP doivent être formées en respectant l'existence d'un minimum de la période CAP afin d'assurer qu'au moins les commandes MAC peuvent encore être transmises. La période CAP doit avoir une taille supérieure ou égale à la constante $aMinCAPLength$ définie par le standard comme étant égale à 440 symboles. Ainsi, la valeur du champ « First CAP Slot » ne pourra jamais prendre la valeur 15 d'où son utilisation pour indiquer le cas où la période CAP occupe toute la période active et que la trame PRTPA ne sera pas transmise.

3.6. Exemple de déroulement du protocole FF-MAC

Nous allons illustrer dans cette section le fonctionnement de notre protocole FF-MAC par un réseau constitué de 5 nœuds, un coordinateur et quatre nœuds terminaux (voir figure 15). Nous supposons deux nœuds terminaux (1 et 2) qui collectent des informations et les envoient aux autres nœuds terminaux (3 et 4 respectivement) pour réaliser une action spécifique (actuateurs par exemple).

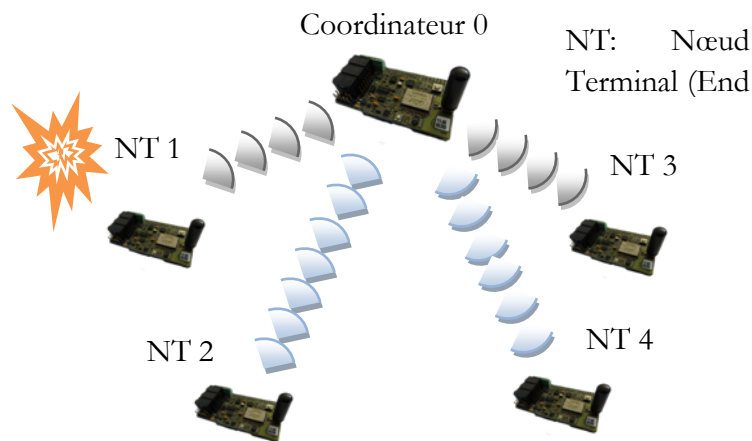


Figure 15 : Exemple d'une topologie de capteurs/actuateurs

Supposons que le nœud terminal « NT 1 » détecte un événement critique, et devra envoyer les informations collectées en temps-réel au nœud « NT 3 ». Tout d'abord, il commencera par l'envoi d'une demande de réservation de GTS auprès du coordinateur. Une fois cette demande est honorée, le nœud peut commencer à utiliser son GTS comme illustré dans diagramme de séquence de la figure 16.

Après la réception du Beacon qui indique le succès de la réservation du GTS pour ce nœud, la période CFP démarre et le nœud « NT 1 » commence l'utilisation de son propre GTS en transmettant directement les paquets à son coordinateur (sans backoff ni CCA). La durée du GTS est supposée suffisante, car c'est le nœud qui l'a choisi selon ses besoins. Cependant, cette durée pourra être plus grande que la durée nécessaire. Cela vient des restrictions du standard qui réserve les GTS par multiple de slots. Ceci introduit en général un gaspillage de bande passante réservée et non utilisée. La solution proposée dans la section 3.7 peut pallier ce problème.

À la fin de la CFP, qui correspond dans notre exemple à la fin du GTS (un seul GTS), le coordinateur commence par vérifier s'il a bien reçu des données à caractère temps-réel (c.-à-d. reçues pendant la période CFP). Dans cet exemple, le nœud « NT 1 » a transmis deux paquets temps-réel pour le nœud « NT 3 ». En conséquence, le coordinateur construit un paquet PRTPA de la manière suivante : le champ « D-GTS count » dans le champ contrôle de la trame doit contenir la valeur 1 pour un seul D-GTS dans la D-CFP. Ensuite, dans la charge de cette trame, le descripteur de ce D-GTS contiendra l'adresse courte du nœud destinataire (NT 3) et la durée suffisante du D-GTS pour transmettre les paquets temps-réels en attente en plus de leurs acquittements s'ils sont requis. Cette durée est donnée en nombre de backoffs au lieu du nombre de slots. Une fois la trame PRTPA est reçue, les nœuds « NT 1 », « NT 2 » et « NT 4 » n'ayant aucune entrée pour eux dans la trame PRTPA, se mettront en veille pour économiser de l'énergie.

Les nœuds qui ont des données à envoyer et/ou des données en attente ayant été annoncées dans le Beacon doivent se réveiller au début de la CAP afin de les échanger avec le coordinateur. Dans ce cas de figure, l'algorithme CSMA-CA est utilisé pour accéder au médium. Les autres nœuds peuvent se mettre en veille jusqu'au début de la supertrame suivante. Le nœud « NT 3 » par contre, changera l'état de son émetteur-récepteur en mode réception pendant toute la durée de son D-GTS afin de pouvoir récupérer les données temps réel du coordinateur.

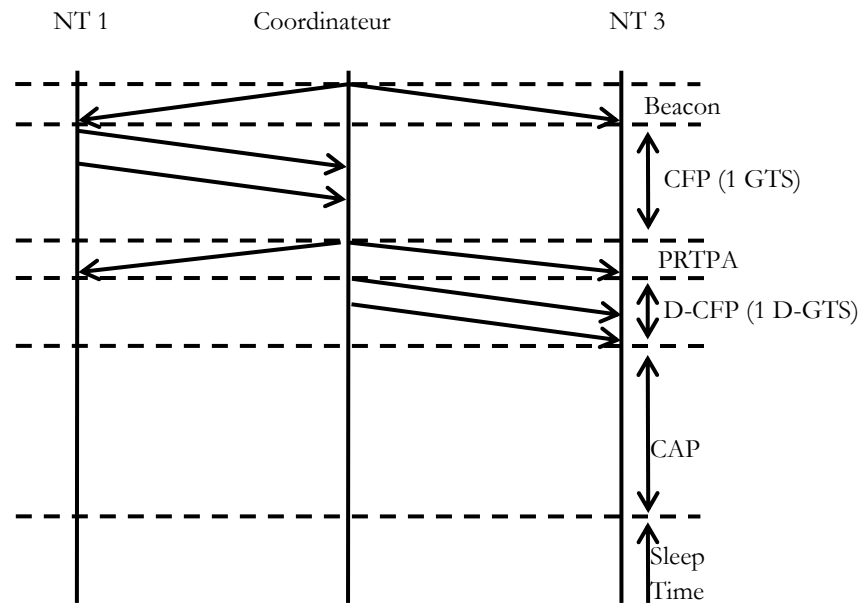


Figure 16 : Exemple d'échange des paquets temps-réel selon le protocole FF-MAC

Ainsi, l'algorithme peut assurer la retransmission en temps-réel des données ayant des contraintes temporelles envoyées du nœud « NT 1 » vers le nœud « NT 3 ». Cette démarche continuera à se répéter de la même manière dans les supertrames suivantes, avec un éventuel changement dans la structure de la supertrame causée par l'ajout ou la suppression d'un GTS, ou par le changement de la quantité de données temps-réel stockées dans la file d'attente du coordinateur.

3.7. Maintenance de la supertrame

Vu les conditions imposées pour préserver la période CAP. La taille de la période D-CFP nécessaire pour offrir une meilleure QoS peut ne pas être honorée. Dans ce cas, le coordinateur peut procéder à une distribution du temps possible selon la quantité des données en attente pour chaque nœud. Chaque candidat pour la D-CFP aura un pourcentage du temps disponible. Ainsi, le coordinateur peut partager la durée de la période D-CFP de manière équilibrée. Certes, ceci

entraînera une dégradation de la qualité de service pour les données temps-réel à cause des ressources limitées. Ce problème peut apparaître généralement quand le réseau n'est pas bien dimensionné. Autrement dit, quand le choix des paramètres réseaux, notamment le paramètre SO, ne prend pas en considération la densité des nœuds ainsi que la quantité estimée du trafic. Une solution intéressante proposée dans [80], qui consiste à augmenter le nombre de slots dans la supertrame, qui est actuellement limité à 16. Dans nos perspectives, nous pouvons joindre à cette proposition la réduction de la valeur de l'ordre de supertrame « SO ». Cela aura comme résultat, une augmentation de la durée de la période active avec une utilisation plus efficace des GTS (slots plus petit). En effet, selon [102], les grandes valeurs de SO (supérieures à 5) introduisent un grand gaspillage de bande passante. Ainsi, cette solution convient parfaitement à notre protocole, car l'augmentation de la taille de la supertrame n'affectera pas le délai pour les paquets temps-réel et la diminution de la valeur de SO apportera une amélioration de l'efficacité d'utilisation de la bande passante dans la période CFP.

3.8. Tolérance aux pertes des paquets PRTPA

Les technologies sans fil offrent une communication moins fiable que celle offerte par les connexions filaires. Ce problème s'aggrave dans le cadre des réseaux de capteurs sans fil qui souffrent d'un taux de perte de paquets plus élevé. Cette perte de paquets peut toucher des trames très importantes, notamment le Beacon et le PRTPA. Dans le protocole FF-MAC, la réception de la trame PRTPA dans chaque supertrame est très importante quand la période CFP existe, car sa perte par un nœud entraîne une perte d'information sur la structure de la période D-CFP. Sachant que cette période est caractérisée par son aspect dynamique, elle peut changer de forme ou disparaître d'une supertrame à l'autre. Pour cette raison, le nœud qui manque la réception de la PRTPA doit suivre la procédure suivante :

- Premièrement, s'il possédait un D-GTS dans la supertrame précédente, le nœud ne doit pas l'utiliser, car il peut y avoir des changements dans la structure de la période D-CFP.
- Deuxièmement, ce nœud n'ayant aucune idée sur le nouvel ordonnancement, ne doit pas commencer la période CAP immédiatement. Il doit supposer le cas le plus défavorable dans lequel la période CAP se réduit à sa taille minimale. Par conséquent, il doit différer ses envois et/ou ses demandes de données en attente jusqu'à $aMinCAPLength$ symboles avant la fin de la période active. Ces mesures ont pour objectif de protéger les D-GTS contre les éventuelles collisions.

3.9. Protocole FF-MAC et les autres approches

Comme nous l'avons présenté dans ce chapitre, notre approche est basée essentiellement sur l'amélioration de la supertrame par changement de l'ordre des périodes d'activité, ainsi que par l'introduction d'une période déterministe pour assurer la retransmission des paquets temps-réel dans la même supertrame. La majorité des approches proposées dans la littérature pour améliorer le standard IEEE 802.15.4 se sont concentrées sur l'amélioration de l'efficacité dans l'utilisation des GTS et l'accélération du processus de réservation. Ces solutions proposées se basent sur des mécanismes qui sont complémentaires au nôtre. Ainsi, on peut facilement les adapter et les fusionner avec le protocole FF-MAC pour plus d'efficacité.

De plus, le groupe de travail TG4e de l'organisme IEEE a récemment proposé une nouvelle version de ce standard nommé IEEE 802.15.4e. Ce dernier a été introduit comme un amendement à la couche MAC de la version 2011 et dont est de fournir un meilleur support dans le marché industriel. La version « e » ne supprime pas son prédécesseur, mais lui ajoute de nouvelles fonctionnalités et de nouvelles formes optionnelles de supertrames. Le choix parmi ces supertrames se fait selon les besoins des applications. Deux structures de supertrames correspondent au contexte de notre étude : la supertrame actuelle du standard 802.15.4-2006 et la DSME multi-supertrame. Ces deux formes de supertrames sont plutôt adaptées aux applications dans lesquelles deux types de trafics homogènes peuvent coexister : un trafic non prioritaire et un trafic temps-réel. Comme perspective, nous pensons que l'adaptation de notre approche à la deuxième forme proposée par le nouveau standard apportera des améliorations importantes.

4. Conclusion du chapitre

Nous avons montré dans ce chapitre que le format actuel de la supertrame présente des limitations qui peuvent dégrader la qualité de service dans le réseau, notamment pour les applications dans lesquelles le trafic temps-réel peut exister. Ces limitations liées à l'ordre des périodes actives (CAP et CFP) ainsi que la discontinuité de la garantie de qualité de service ont été corrigées par la proposition d'une structure améliorée de la supertrame. Ce nouveau format permet de transmettre les paquets de la source à la destination dans la même supertrame de manière déterministe. Cette possibilité nous permet de garantir un délai fixe et réduit, même dans des réseaux à faible rapport cyclique (Duty Cycle). Ainsi nous avons proposé une approche efficace pour résoudre le compromis énergie/délai. Une approche qui peut facilement être fusionnée avec d'autres approches proposées dans la littérature pour améliorer le standard IEEE 802.15.4 du fait que leurs principes sont complémentaires.

Jusqu'à présent nous avons présenté le protocole d'une manière fonctionnelle et nous avons montré comment ce protocole pourra améliorer la qualité de service par rapport au standard IEEE 802.15.4. Cependant, une évaluation de performance est nécessaire pour prouver les résultats attendus. L'évaluation de performance qui sera présentée dans le chapitre suivant a été réalisée selon deux approches. D'une part, une évaluation de performance plutôt théorique a été réalisée par simulation dans l'environnement de simulation NS-2 et d'autre part, une validation du protocole sur des nœuds RSCF réels a été effectuée afin de prouver les améliorations d'un point de vue pratique.

CHAPITRE III :

Évaluation des Performances du Protocole FF-MAC par Simulation et Expérimentation

Le présent chapitre est dédié à la discussion des différents scénarios de tests que nous avons effectués afin d'évaluer les performances de notre approche FF-MAC par rapport au mécanisme des GTS proposé dans le standard IEEE 802.15.4.

Les différents scénarios sont choisis, d'une part, pour mettre en évidence et prouver les limitations que nous avons identifiées et d'autre part, pour montrer l'efficacité des améliorations apportées par le protocole FF-MAC.

L'évaluation des performances est réalisée par deux méthodes complémentaires qui constituent les deux parties principales de ce chapitre. La première est consacrée aux travaux de simulation sous le simulateur NS-2 et la deuxième présente les résultats obtenus par prototypage sur des nœuds RCSF réels.

1. Évaluation des performances par simulation

1.1. Présentation du simulateur NS-2

NS-2 [103] « Network Simulator version 2 » est un outil de simulation « Open Source » qui appartient à la famille des simulateurs événementiel « Event-Driven » [104]. Ce simulateur a été conçu pour la recherche spécialisée dans le domaine des réseaux informatiques, plus particulièrement, les protocoles de communication filaire et sans-fil¹¹. NS-2 est l'un des simulateurs les plus utilisés dans ce domaine de recherche. Il dispose d'un nombre important de modèles¹² de protocoles de toutes les couches du modèle OSI « Open System Interconnexion » allant de la couche physique à la couche Application.

Comme les autres simulateurs de réseaux, NS-2 a été développé en utilisant deux langages de programmation : le langage C++, qui définit les mécanismes internes des objets de simulation (le noyau du simulateur) et le langage Otcl « Object-oriented Tool Command Language » qui est responsable de la configuration et la liaison de ces objets ainsi que l'ordonnancement des événements « Events Scheduling ». Il joue le rôle d'une interface entre l'utilisateur et le noyau du simulateur à travers le script TCL « Tool Command Language ». Cette architecture, illustrée dans la figure 17, est introduite afin de permettre une exécution rapide grâce au langage C++ pour le détail d'implémentation des protocoles, en plus de la possibilité de modifier la configuration de manière rapide et interactive à travers le langage OTCL.

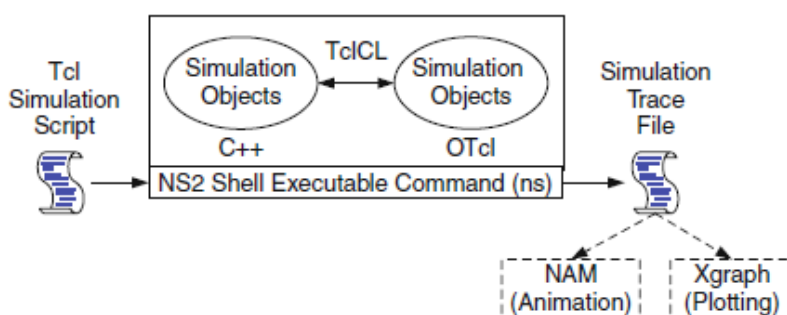


Figure 17 : Architecture de base de NS-2 (source [104])

¹¹ Au début, le simulateur NS-2 a été conçu pour les réseaux filaires. La composante sans fil a été ajoutée ultérieurement en tant qu'extension au simulateur.

¹² La version officielle dispose de plusieurs modèles de protocole. Cependant, elle ne n'inclue pas tous les modèles de simulation existants. Ces derniers peuvent être téléchargés et inclus manuellement. Une liste non exhaustive des ces modèles est disponible dans [105].

1.2. Modèle WPAN

Le simulateur NS-2 dispose d'un modèle mature pour le standard IEEE 802.15.4 nommé « WPAN » [106]. Il s'agit d'un des modèles du standard IEEE 802.15.4 les plus utilisés par les chercheurs dans le domaine des RCSF. Ce modèle dispose d'une implémentation de la couche physique (PHY 2.4 GHz), la sous-couche MAC et la sous-couche de convergence SSCS « Service-Specific Convergence Sub-layer ». En ce qui concerne la sous-couche MAC, le modèle WPAN supporte les modes non beacon-enabled (asynchrone) et beacon-enabled (synchrone). Il supporte aussi les différentes topologies définies dans le standard, à savoir : la topologie en étoile, la topologie point-à-point et la topologie en arbre de « clusters » (cluster tree). Cependant, le mode avec beacon n'est pas complet, car lui manque le mécanisme de gestion des GTS. Or nos travaux visent l'amélioration du standard IEEE 802.15.4 par une gestion appropriée de la supertrame centrée sur l'ordonnancement des périodes CAP et CFP 'GTS'.

Dans notre étude de simulation, nous avons utilisé ce modèle et nous l'avons complété par un module de gestion des GTS pour qu'on puisse comparer notre approche avec le standard. Ce module contient :

- Un gestionnaire des demandes de GTS.
- Une base de données des GTS, avec une gestion des allocations.
- La gestion et l'utilisation des slots alloués.

1.3. Avantages et limites de la simulation

L'utilisation de la simulation présente plusieurs avantages. D'une part, la simulation est peu coûteuse, car les simulateurs existants sont majoritairement gratuits et Open Source. D'autre part, le temps de développement et de simulation est très inférieur comparé aux implémentations réelles. En outre, le simulateur permet d'accélérer le temps et de prédire le comportement du système étudié. À titre d'exemple, on peut simuler la durée de vie des capteurs en un temps très court, vu que le temps lui-même est virtuel. Aussi, le simulateur peut être un outil d'évaluation complémentaire aux expérimentations réelles, car dans certains cas de figure, il est difficile d'avoir un grand nombre de nœuds réels, d'où l'intérêt de la simulation qui permet de faire le passage à l'échelle par une simple modification du script de simulation.

On parle dernièrement d'un nouveau paradigme dans lequel le simulateur ne joue pas seulement le rôle d'un outil d'évaluation, mais aussi sera une composante qui participe de manière

active à la réalisation de quelques tâches dans le système réel. Par exemple, en se basant sur les résultats obtenus du réseau réel, le simulateur peut créer les tables de routage et distribuer les changements aux nœuds de ce réseau [107].

Malgré tous ces avantages, les outils de simulation présentent quelques limites. Le plus grand inconvénient étant lié aux modèles des protocoles et leurs degrés de correspondance à la réalité (p. ex. propagation des ondes radio). En effet, le modèle utilisé pour simuler un protocole est un modèle théorique qui a recours à plusieurs simplifications. Notamment, dans notre cas, la couche physique et le modèle de propagation sont très simples et n'introduisent pas toutes les imperfections du média réel. Une étude a été menée dans [108] dans l'optique de voir la pertinence des résultats de simulation comparés aux implémentations réelles dans les réseaux sans fil. Cette étude montre que les résultats de simulation peuvent être proches des résultats réels dans certaines conditions, surtout si le choix du modèle de la couche physique et de ses paramètres (plus particulièrement, le modèle de propagation) est correct. Mais, le simulateur peut ne pas être adapté dans d'autres contextes. Selon Shannon [109], la simulation est « le processus de concevoir un modèle pour un système réel et de réaliser des expériences avec ce modèle dans l'optique de comprendre le comportement de ce système et/ou d'évaluer ses différentes stratégies opérationnels ». Ainsi, les résultats obtenus par simulation peuvent donner une estimation des performances, mais chaque protocole doit être validé sur des équipements réels afin d'obtenir des résultats plus précis et plus réalistes.

1.4. Paramètres de simulation

Dans l'évaluation des performances, nous allons comparer trois approches, à savoir : le mécanisme de GTS selon le standard IEEE 802.15.4, la forme améliorée de la supertrame [100] et le protocole FF-MAC [101].

L'évaluation de performance a été réalisée sur le modèle WPAN du simulateur NS-2 (version 2.34). Les simulations adoptent les suppositions suivantes : la couche MAC du standard IEEE 802.15.4 opère en mode « Beacon-enabled », tous les paquets unicast envoyés doivent être acquittés par un acquittement de niveau MAC et le protocole ARP est désactivé. La charge du trafic est variée par le changement du nombre des nœuds du réseau. Dans tous les scénarios, le réseau est constitué d'un coordinateur du PAN placé au centre du réseau en étoile, entouré par un nombre variable de nœuds distribué aléatoirement et a porté radio du coordinateur. Le routage n'est pas utilisé, car d'une part, nous utilisons un réseau en étoile géré par un coordinateur. Et d'autre part, nous souhaitons évaluer les performances sans influence des

couches supérieures. Au niveau de la couche Application, tous les nœuds transmettent des paquets UDP avec une fréquence d'un paquet chaque BI (Beacon Interval), car dans les réseaux réels, le paramètre BO peut être choisi selon la fréquence d'échantillonnage souhaitée. Un exemple d'un réseau de 51 nœuds est illustré à la figure 18.

Le tableau 5 résume les différents paramètres utilisés dans la simulation.

Paramètres	Valeur
Nombre de nœuds	10, 20, 30, 40, 50, 60, 70, 80
Topologie	Étoile
Taille des paquets	50 Octets
Simulation time	2000 secondes
Transmission range	15m
Transmission rate	1 paquet / BI
Application	UDP client/serveur
protocole de routage	Aucun
ARP	désactivé
PHY	PHY 2.4 GHz / 250 kbps
Modèle de propagation	Two-ray ground reflection model [110]
Nombre de scénarios	Plus que 110

Tableau 5 : Résumé des paramètres de simulation

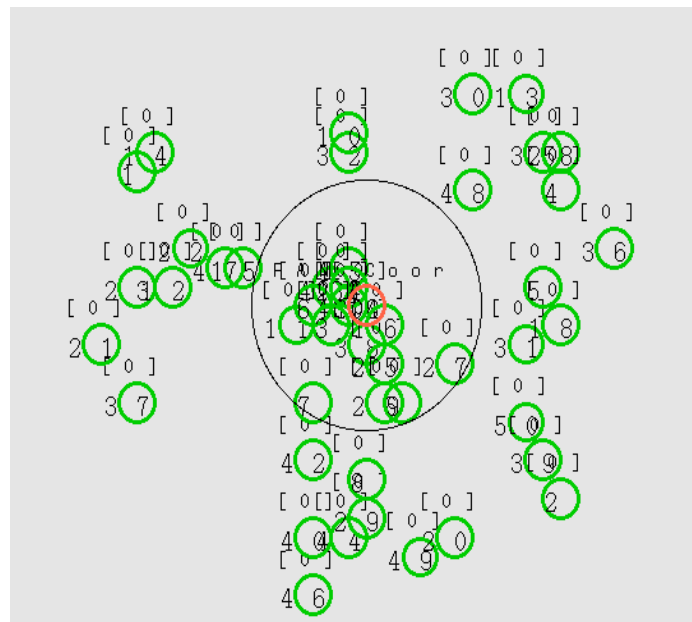


Figure 18 : Exemple d'un réseau de 51 nœuds visualisé à l'aide de NAM [111]

1.5. Résultats obtenus

La figure 19 montre l'évolution du délai de bout en bout pour différentes valeurs du paramètre BO « Beacon Order ». Le réseau est constitué de 21 nœuds. Nous pouvons constater que le format amélioré de la supertrame offre des performances meilleures que celles du standard IEEE 802.15.4. Le délai est réduit d'environ 45% grâce à l'amélioration de la supertrame. Cependant, le délai augmente considérablement dans les deux protocoles avec l'augmentation du temps de sommeil. En effet, les deux protocoles utilisent la méthode d'accès CSMA-CA pour la réception des paquets. L'aléa inhérent à cet algorithme rend la réception des paquets par leurs destinations non déterministes et peut augmenter le délai à des valeurs très importantes. Ainsi, le format amélioré de la supertrame permet la retransmission des données dans la même supertrame ce qui justifie les valeurs du délai de communication inférieure à celles du standard. Cependant, le délai reste sensible au temps de sommeil, car il n'y a aucune garantie de retransmission. Dans la même figure 19, nous constatons que le protocole FF-MAC a résolu ce problème grâce à l'ordonnancement déterministe introduit avec la période D-CFP. Les délais mesurés sont très faibles avec une amélioration qui peut atteindre les 90 % par rapport à celle du standard IEEE 802.15.4. Aussi, grâce à la transmission déterministe de bout en bout, le protocole FF-MAC nous permet d'estimer des bornes de délai pour certains scénarios.

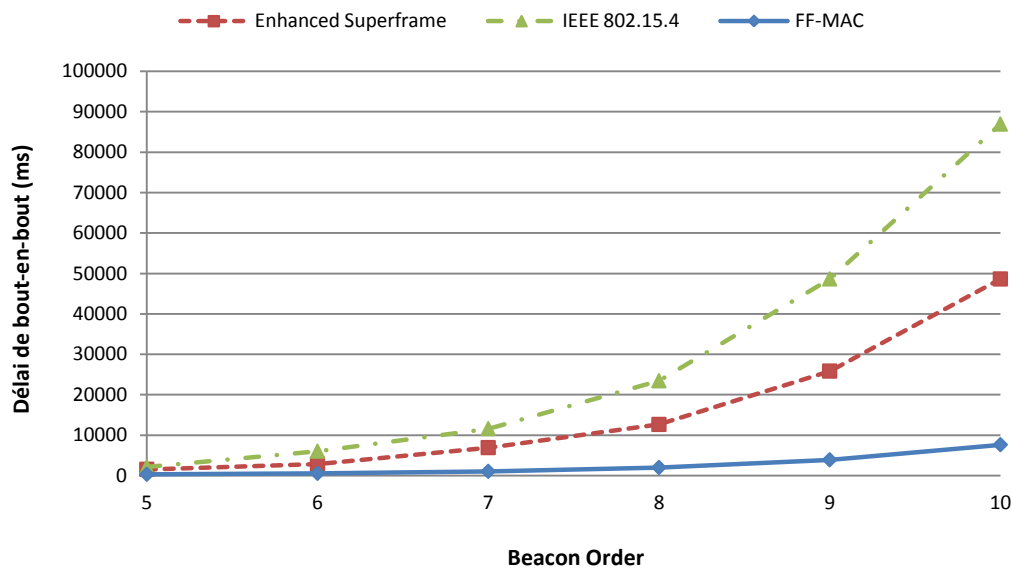


Figure 19 : Délai de bout en bout en fonction du paramètre BO (21 nœuds, SO=5)

Dans la figure 20, nous avons augmenté le nombre de nœuds du réseau à 31 nœuds pour évaluer son impact sur le délai de bout en bout. Les résultats obtenus montrent que le délai de bout en bout a augmenté dans le cas du standard et du format amélioré avec l'augmentation de la

densité de nœuds. Cela vient de l'utilisation du mécanisme CSMA-CA dans les communications descendantes (c.-à-d. du coordinateur vers la destination finale). Cependant, le format amélioré a réduit le délai de bout en bout moyen d'environ 35 % par rapport au standard, car ce format offre la possibilité de relayer les données temps-réel dans la même supertrame, même sans garantir leurs arrivées à leurs destinations finales. Par contre, le protocole FF-MAC n'est affecté ni par le changement en nombre de nœuds ni par l'augmentation de la durée de la supertrame, car les données temps-réel sont transmises de manière déterministe de bout-en-bout à l'intérieur de la même supertrame.

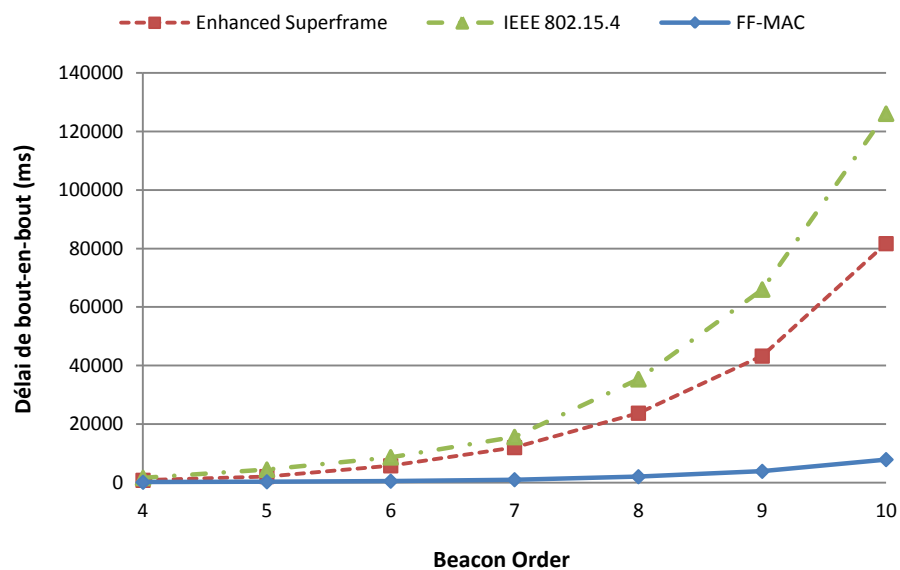


Figure 20 : Délai de bout-en-bout en fonction du BO « Beacon Order », avec (N=31 et SO=4)

La figure 21 montre le comportement du délai mesuré pour différents rapports cycliques (Duty Cycle). Les résultats confirment que les performances du protocole FF-MAC ne sont pas affectées par les longues périodes de sommeil. Ainsi, FF-MAC est très adapté aux réseaux peu actifs ayant des données à forte contrainte temporelle (prioritaires).

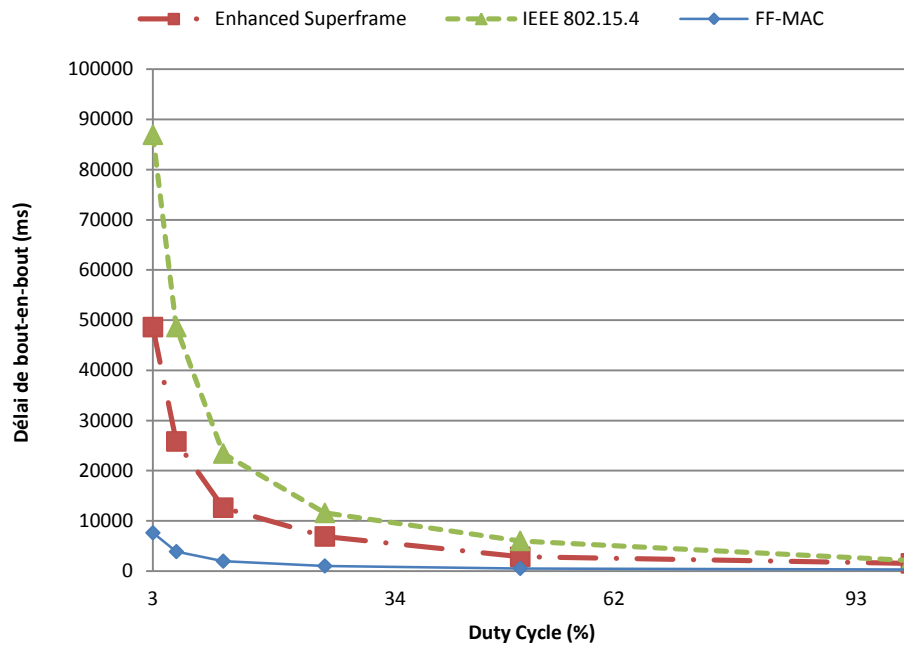


Figure 21 : Délai de bout-en-bout en fonction du rapport cyclique pour un réseau de 21 nœuds

Plus de scénarios ont été réalisés pour évaluer l'impact du nombre de nœuds du réseau sur le délai de communication. Les valeurs des paramètres BO et SO sont configurées pour un faible rapport cyclique (BO=10, SO=4, DC = 1.56%). Les résultats de la figure 23 présentent une stabilité et des valeurs faibles du délai mesuré des données temps réel pour le protocole FF-MAC. Les autres protocoles sont très sensibles à ces variations.

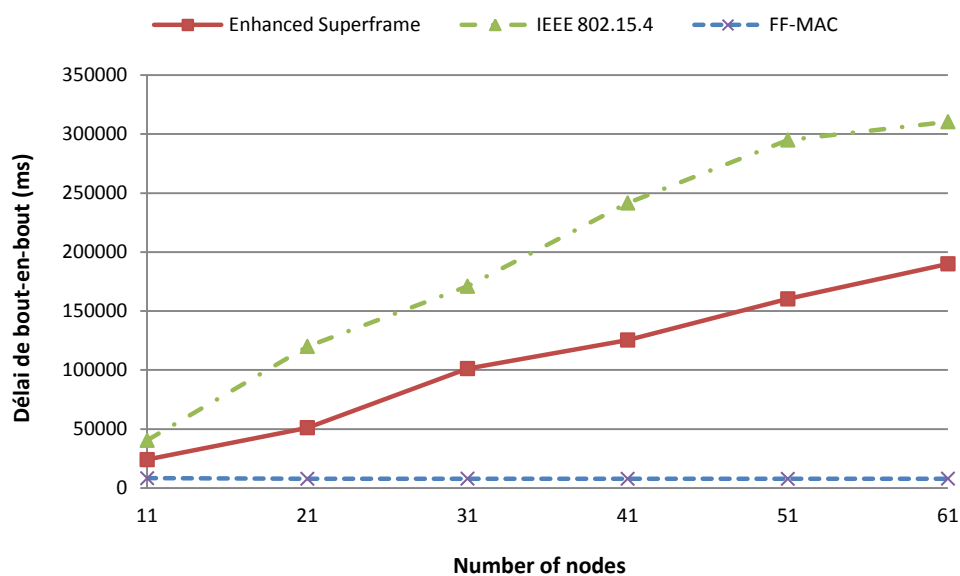


Figure 22 : Délai de bout en bout pour différentes densités de réseau

Dans la figure 23, nous avons regroupé un grand nombre de scénarios dans lesquels ont été changées à la fois la densité des nœuds et la durée de la supertrame pour évaluer le comportement du protocole FF-MAC en fonction de ces paramètres. Pour toutes les densités choisies (21, 31, 41, 51, 61, 71 et 81 nœuds), le protocole FF-MAC offre un délai très stable et indépendant du nombre de nœuds dans le réseau grâce à l'acheminement déterministe de bout en bout.

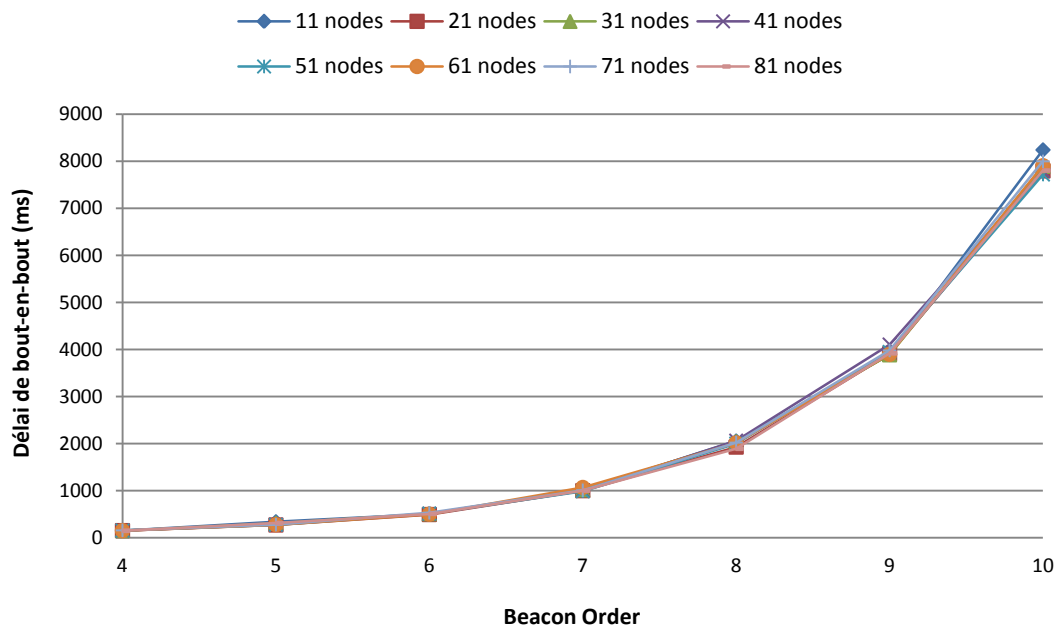


Figure 23 : Délai de bout-en-bout en fonction du paramètre BO pour différentes densités des nœuds

Un résumé d'une partie des scénarios de simulation réalisés pour le protocole FF-MAC est illustré à la figure 24.

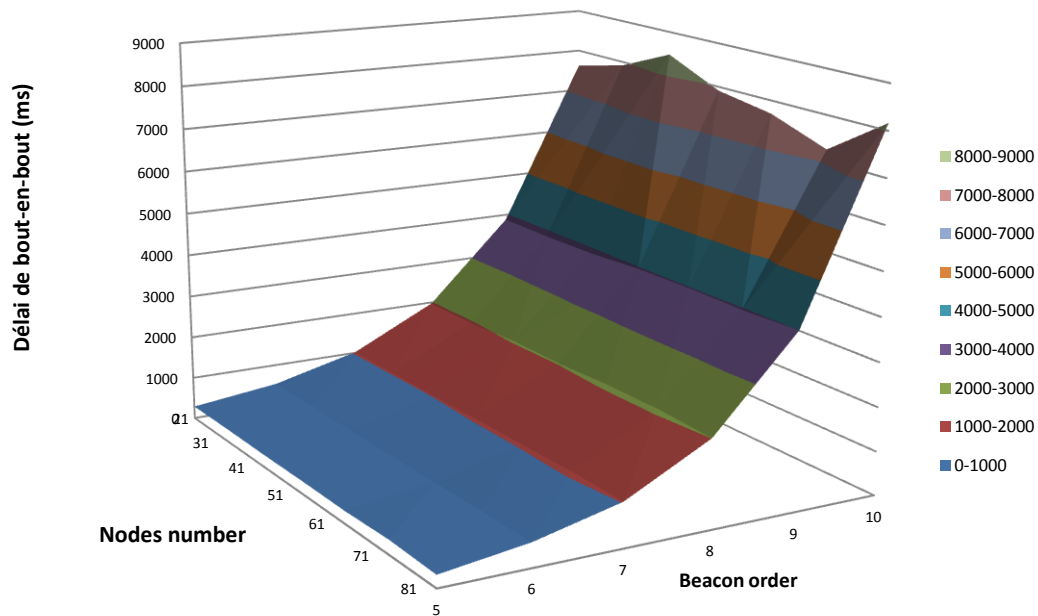


Figure 24 : Résumé des scénarios de simulation du protocole FF-MAC

1.6. Conclusion de la première partie

Les résultats de simulation présentés dans cette partie confirment les améliorations apportées par le protocole FF-MAC grâce à ces nouvelles fonctionnalités. Les scénarios ont été choisis pour vérifier l'efficacité de notre approche en fonction de la densité des nœuds, des paramètres BO et SO qui déterminent la durée de la supertrame et la durée des slots respectivement. Le délai mesuré dans tous ces scénarios reste faible et stable par l'utilisation du protocole FF-MAC.

Les résultats obtenus ont été confirmés dans un environnement plus réaliste sur des plateformes de capteurs réelles. Cette étude expérimentale fera le sujet de la deuxième partie de ce chapitre.

2. Évaluation de performances expérimentale du protocole FF-MAC

2.1. Importance de l'étude expérimentale dans la recherche appliquée

Dans notre domaine de recherche appliquée, les résultats de simulation, malgré leur importance, ne sont pas suffisants. En effet, les modèles de simulation sont beaucoup plus simplistes et en conséquence, ils ne reflètent que partiellement le comportement des systèmes physiques réels (p. ex. interférence). Les réseaux réels, quant à eux, comportent plusieurs facteurs (p. ex. interférence, multi-trajets, etc.) qui peuvent avoir une influence sur les résultats obtenus. Ainsi, la validation de notre approche sur des nœuds RCSF réels « Prototype » et dans un environnement réaliste est nécessaire. Cette environnement caractérisé par une communication peu fiable et un ensemble de contraintes intrinsèques des nœuds RCSF (p. ex. faible capacité de stockage, puissance de calcul limitée, etc.).

Ce prototypage [112] nous a permis de valider le bon fonctionnement du protocole FF-MAC ainsi que les différentes améliorations qu'il apporte d'un point de vue pratique.

2.2. Contraintes d'implémentation

L'implémentation sur des prototypes est un processus long et très contraignant, car les nœuds RCSF réels imposent des contraintes de taille limitée du code, capacité limitée de la mémoire RAM en plus d'une difficulté de débogage sur les capteurs, car il nécessite des outils spécialisés (p. ex. émulateur ...). Par exemple, puisque les nœuds font partie d'un réseau synchronisé par des beacon, l'utilisation des points d'arrêt « break point » dans un nœud entraînera sa déconnexion de son coordinateur à cause de la perte d'un ou plusieurs Beacons. Si ce nœud est propriétaire d'un GTS, l'utilisation de ce dernier est interdite selon les règles du standard IEEE 802.15.4. Pour pouvoir déboguer notre implémentation, nous avons utilisé l'interface UART pour transmettre les informations requises. Cependant, cette technique doit être utilisée avec modération. En effet, les informations envoyées via l'interface série sont des chaînes de caractères. Si la taille de ces dernières est importante, elle risque de déborder la mémoire des capteurs et entraîner son échec. Ce comportement a été constaté dans les nœuds à très faible capacité de stockage (p. ex. la carte MeshBean 2 avec 8 Koctets de mémoire de données). De plus, l'utilisation intensive de l'interface UART consommera des cycles d'horloge (p. ex. pour le formatage des chaînes de caractère, la création du message à transmettre selon le protocole de communication entre le nœud et l'ordinateur, puis la transmission via l'interface UART). Ceci peut ralentir le fonctionnement du système et entraîner des délais supplémentaires. L'obtention

des résultats est aussi plus compliquée, car dans notre cas, la mesure du délai nécessite la connaissance du temps d'envoi et de réception des paquets, alors que les horloges des nœuds RCSF ne sont pas synchronisées. Ainsi, il nous fallait développer un logiciel spécifique pour cet effet qui sera présenté dans le paragraphe 2.3.3.

2.3. Détail technique et outils utilisés dans le prototypage

2.3.1. Présentation de la plateforme de test

Nos travaux d'expérimentation ont été menés essentiellement sur la plateforme iLive, car il s'agit d'une plateforme développée au sein de notre laboratoire par l'équipe SMIR¹³, une plateforme performante et compatible avec le code open source sur lequel nous nous sommes basés dans notre implémentation. En outre, d'autres plateformes, à savoir : STK600-Atmega128RFA1, RCB128RFA1 et Meshbean 2 « ZigBit », ont été utilisées afin de montrer l'indépendance des résultats obtenus des plateformes de nœuds RCSF utilisées. Une partie de ces capteurs est présentée dans la figure 25.

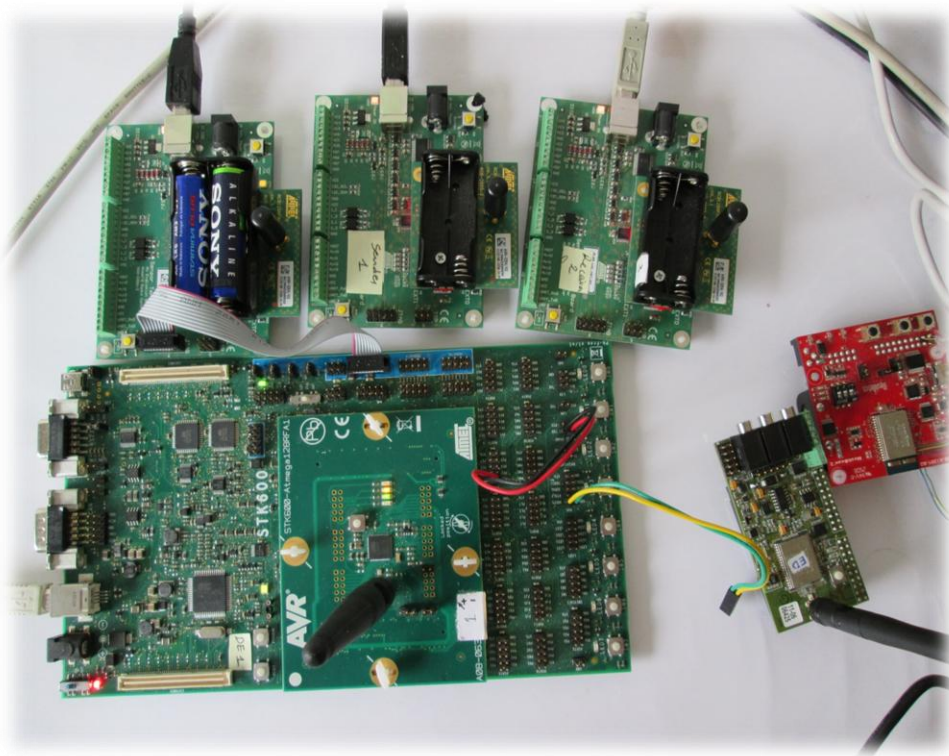


Figure 25 : Matériel utilisé dans les expériences

¹³ Le groupe SMIR du laboratoire LIMOS UMR 6158 CNRS de l'Université Blaise Pascal

2.3.1.1. Caractéristiques des nœuds RCSF utilisés

Le tableau 6 résume quelques caractéristiques importantes des plateformes de test utilisées dans nos travaux expérimentaux.

Plateforme	iLive	STK600-Atmega128RFA1	RCB128RFA1	Meshbean2 (ZigBit)
Couche PHY	PHY 2.4GHz DSSS O-QPSK	PHY 2.4GHz DSSS O-QPSK	PHY 2.4GHz DSSS O-QPSK	PHY 2.4GHz DSSS O-QPSK
Débit binaire (Kbits/s)	250	250 (High data rate support)	250 (High data rate support)	250
Débit de symbole (Ksymboles/s)	62.5	62.5	62.5	62.5
Fréquence Max MCU (MHz)	16	16	16	16
Mémoire App./Boot (Koctets)	128	128	128	128
Mémoire des données (Octets)	8192	16384	16384	8192
EEPROM (Octets)	4096	4096	4096	4096

Tableau 6 : Caractéristiques des plateformes utilisées dans l'évaluation de performances

Remarque : à l'exception des nœuds RCFS basés sur le module radio AT86RF230 (en l'occurrence Meshbean 2 et iLive « 1^{re} génération »), il est possible d'atteindre un débit maximal de 2 Mbits/s en dehors du standard, comme il est indiqué dans le tableau 7 [113].

Frequency band/transceiver	MAC-2006 compliant	High data rate mode 1	High data rate mode 2
Sub-1 GHz Channel 0 Channel 1-10 AT86RF212		Channel Page 16 200kb/s 500kb/s	Channel Page 17 400kb/s 1000kb/s
Chinese Band Channel 0-3 AT86RF212		Channel Page 18 500kb/s	Channel Page 19 1000kb/s
2.4GHz Channel 11-26 AT86RF231, Atmega128RFA1, ...	Channel Page 2 500kb/s	Channel Page 16 1000kb/s	Channel Page 17 2000kb/s

Tableau 7 : Canaux propriétaires d'ATMEL pour les hauts débits binaires non-standard

2.3.2. Pile de protocoles Open Source « Atmel MAC Stack »

Atmel, leader mondial dans le domaine des semi-conducteurs avancés, publie une pile protocolaire « Open Source » appelée « Atmel MAC Software for IEEE 802.15.4 transceivers »

[113]. Ce logiciel supporte les différents microcontrôleurs et plateformes/cartes d'ATMEL. Il permet d'accéder à l'ensemble des fonctionnalités proposées par le standard IEEE 802.15.4-2006. Ainsi, les deux modes de la sous-couche MAC ; « beacon-enabled » et « non beacon-enabled », sont disponibles. L'architecture de cette pile, illustrée dans la figure 26, est constituée de :

- Une couche d'abstraction de la plateforme, PAL « Platform Abstraction Layer »
- Une couche d'abstraction de l'émetteur-récepteur, TAL « Transceiver Abstraction Layer » et le module d'accès aux fonctionnalités des émetteurs-récepteurs, TFA « Transceiver Feature Access »
- Une couche MAC qui englobe le noyau de la couche MAC et l'API MAC.
- Une couche d'abstraction de la sécurité, SAL « Security Abstraction Layer » et la boîte à outils de sécurité, STB « Security Toolbox ».
- Un module de gestion des ressources qui contient le gestionnaire des tampons et celui des files d'attente, Buffer and Queue Management (BMM and QMM)

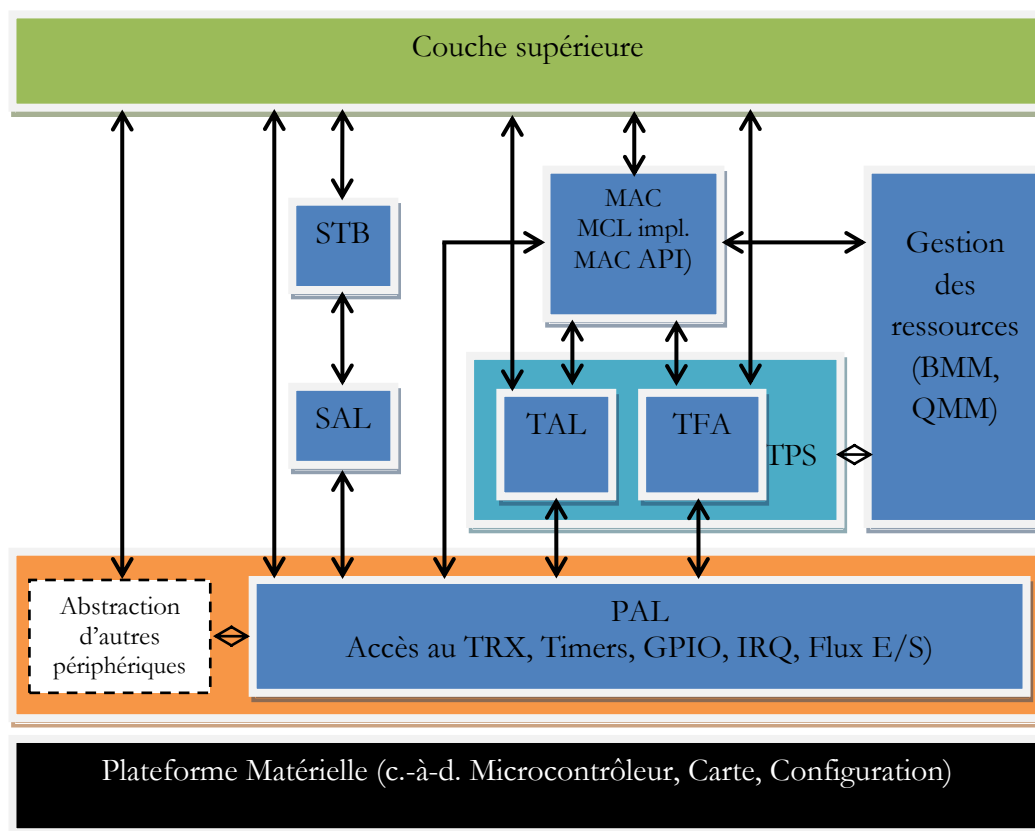


Figure 26 : Architecture du logiciel MAC d'ATMEL pour les nœuds IEEE 802.15.4 [113]

La pile de protocoles open source d'Atmel est modulaire, ce qui permet à l'utilisateur de choisir les modules dont il a besoin. L'interaction entre les différentes composantes doit suivre les directives indiquées par des flèches dans la figure ci-dessus.

Ce package est développé selon une approche générique et n'est pas optimisé pour les besoins d'une application spécifique. Cela permet à l'utilisateur de l'adapter plus facilement à ses propres applications et de l'étendre par d'autres protocoles de niveau supérieur, comme les protocoles de routage. Notons que l'implémentation favorise l'optimisation des ressources et la modularité sur le respect du modèle OSI, notamment en ce qui concerne l'abstraction des couches. Une partie de la couche MAC est déplacée vers la couche physique (p. ex. l'algorithme CSMA-CA).

La version actuelle de ce logiciel (version 2.8.0) ne prend pas en charge le mécanisme des GTS. Ainsi, étant primordiale dans notre étude d'une part, pour implémenter notre protocole FF-MAC basé le standard IEEE 802.15.4 et d'autre part, pour comparer les performances de notre protocole à ceux de ce dernier, nous avons complété cette pile protocolaire par l'ajout des composantes indispensables à l'utilisation des GTS. Nous avons ajouté un module qui couvre principalement les fonctionnalités suivantes :

- La gestion des commandes de demande de GTS « GTS_Request ».
- La gestion de la base de données des GTS ainsi que l'annonce des changements dans le Beacon.
- La gestion et l'utilisation des slots alloués.

2.3.3. Le logiciel JSensorsMonitor

Pour prendre des mesures sur les prototypes développés et d'en extraire des résultats, nous avons développé un outil qu'on a appelé : « JSensorsMonitor » (voir figure 27). Cet outil permet de réaliser des tâches de supervision d'un réseau de capteurs sans fil. Il permet par exemple de stocker les données échantillonnées par les nœuds RSCF, visualiser la topologie et suivre en temps-réel l'évolution de certains paramètres et données échantillonnées. En plus, ce logiciel est doté d'autres fonctionnalités utiles pour l'implémentation, le débogage et la validation du bon fonctionnement du protocole développé. JSensorsMonitor peut aussi jouer le rôle d'un sniffer logiciel lorsqu'il est couplé à un nœud RSCF configuré en mode « promiscuous ». Il peut ainsi générer des fichiers au format PCAP qui peuvent être analysés ultérieurement par les outils

d'analyse des paquets existants (p. ex. Wireshark et tcpdump). L'architecture de ce logiciel sera présentée dans l'annexe.

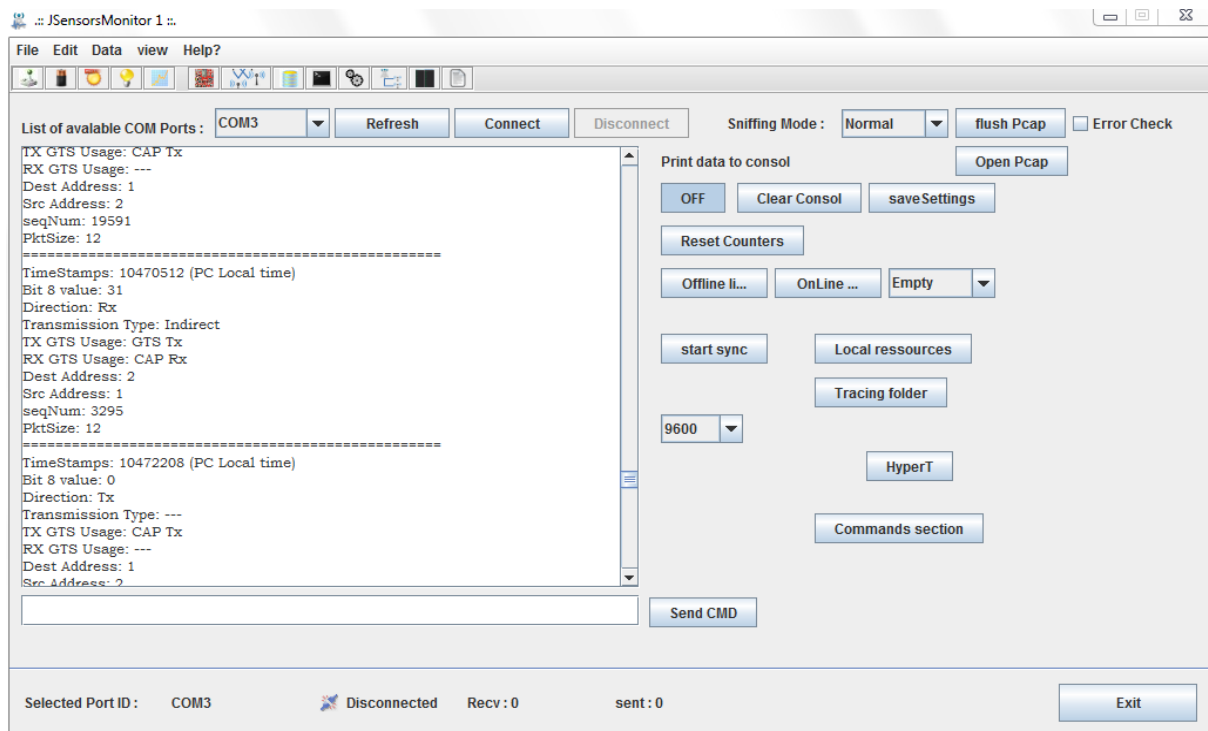


Figure 27 : Interface principale du logiciel « *JSensorsMonitor* »

2.4. Implémentation du protocole FF-MAC

Comme nous l'avons mentionné dans les paragraphes précédents, notre couche MAC est développée sur la base de la pile Open Source d'Atmel « Atmel MAC Stack ». Avant l'implémentation du protocole FF-MAC, nous avons complété ce logiciel par l'ajout des mécanismes de gestion des GTS.

L'implémentation du protocole FF-MAC a été réalisée selon trois étapes. Dans un premier temps, nous avons modifié la structure de la supertrame définie dans le standard IEEE 802.15.4-2006 en adoptant le format amélioré. Dans un second temps, nous avons procédé à la modification des trames existantes concernées et la création des nouvelles trames. En effet, nous avons modifié une partie de la trame beacon pour que la valeur du champ « Final CAP Slot » indique le début de la période CAP plutôt que sa fin. Il est à noter que nous n'avons introduit aucun changement dans le format du beacon, mais plutôt dans la signification de ce champ ce qui permet de garder un minimum de compatibilité avec la norme originale (voir le paragraphe 2.6). Ensuite, nous avons développé le module permettant de créer, de transmettre

puis de traiter la trame PRTPA à sa réception. Finalement, nous avons développé le module permettant de gérer et d'utiliser les différentes périodes CFP, D-CFP et CAP.

Remarque : nous avons décrit la trame PRTPA dans le chapitre II comme étant une nouvelle trame MAC. Cependant, dans l'implémentation réelle, nous avons traité la trame PRTPA comme une trame de commande MAC. Ceci nous a permis de simplifier l'implémentation et de réutiliser une partie du code existant pour le traitement des commandes MAC. Afin de distinguer cette trame des autres commandes MAC, nous avons utilisé le premier identifiant réservé disponible « 0x0a ».

2.5. Expérimentations

2.5.1. Schéma expérimental

Nous avons implémenté les deux protocoles, FF-MAC et le mécanisme de gestion des GTS du standard IEEE 802.15.4, dans la pile de protocoles « Atmel IEEE 802.15.4 MAC software package ». Ces programmes « firmware » ont été utilisés dans les nœuds RCSF pour réaliser ces travaux expérimentaux.

Pour mesurer le délai de communication, il est possible de mesurer la durée entre la transmission d'un paquet et la réception de son acquittement transmis par la destination finale. Ainsi, le délai peut être calculé comme étant la moitié de cette durée. Cependant, cette façon de calcul n'est pas adaptée à nos scénarios de test, car la communication dans les deux sens (aller et retour) n'est pas symétrique. Donc pour faire des mesures plus précises, nous avons utilisé le logiciel « JSensorsMonitor » pour collecter les traces de façon similaire à celle adopté par les simulateurs de réseaux. En effet, plusieurs nœuds peuvent se connecter à ce logiciel pour que ce dernier puisse collecter les informations échangées. Ensuite, le logiciel sauvegarde ces données dans une base donnée pour un traitement ultérieur. Toutes les informations sont horodatées par l'application, ce qui permet d'avoir une horloge de référence unique. Ainsi, la suite des calculs des différentes métriques est similaire à celle des simulateurs. Le logiciel « JSensorsMonitor » est doté d'un convertisseur qui permet de convertir les informations sauvegardées dans la base de données à un format similaire aux traces générées par les simulateurs NS-3. Cela nous a permis de réutiliser les mêmes outils et scripts de calcul de ce dernier. Par exemple, pour mesurer le délai entre deux nœuds, nous avons connecté l'émetteur et le récepteur à un ordinateur à travers une liaison série (port USB ou port série « COM ») comme il est indiqué dans la figure 28. Ainsi, le logiciel JSensorsMonitor installé dans cet ordinateur est notifié de chaque transmission ou réception des paquets par les nœuds sources et destination. Pour simplifier les mesures, nous

n'avons pris en compte que les paquets au niveau de la couche application. Mais le processus est identique pour les autres niveaux.

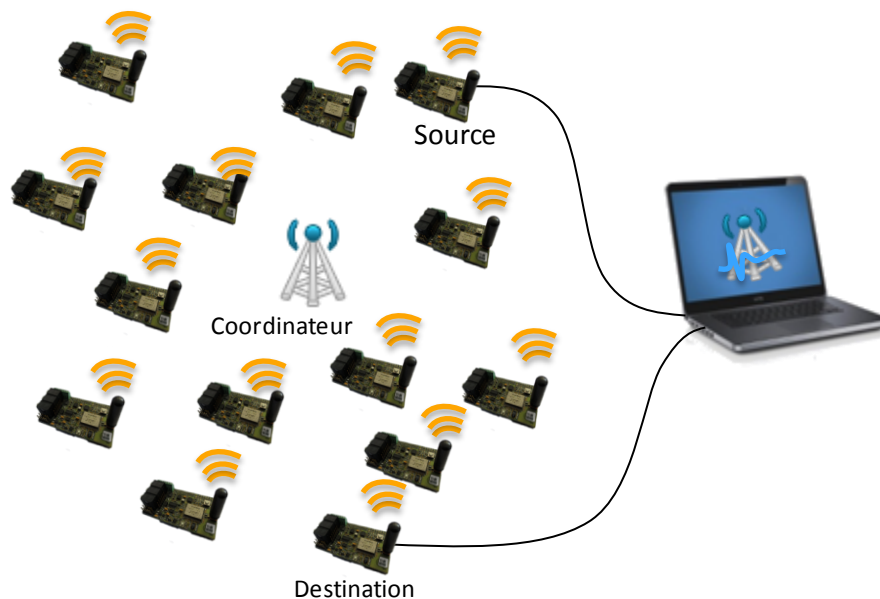


Figure 28 : Schéma de mesures

Les paquets sont marqués par des labels de 32-bits (numéro de séquence) dans la charge de la couche application. Ces labels permettent d'identifier facilement les paquets dans les traces. L'acquittement est activé dans la sous-couche MAC afin de fiabiliser la liaison sans fil (trois retransmissions au maximum).

Remarque : les données reçues par ce logiciel sont horodatées, en utilisant l'horloge de la machine virtuelle, avec une résolution de l'ordre du nanoseconde à l'aide de la méthode prédéfinie de Java « `System.nanoTime()` ». Cela signifie qu'on peut avoir une erreur d'arrondissement au maximum de 0.5 nanoseconde.

2.5.2. Paramétrage des scénarios

Dans toutes les expériences, les nœuds RCSF transmettent dans la bande de fréquences 2.4 GHz avec un débit binaire de 250 kb/s. Chaque nœud transmet un paquet de 12 octets de charge utile chaque BI. Le délai mesuré représente la durée entre la création du paquet dans la couche application du nœud source et sa réception par la destination finale (de bout en bout). Ce délai inclut le délai d'accès au médium, le délai de propagation et la latence due au traitement des trames dans les nœuds intermédiaires (le coordinateur dans notre cas). L'environnement de test, à l'intérieur du bâtiment du laboratoire, est proche des applications « indoor ». Cet

environnement contient des obstacles fixes (ordinateurs, chaises ...) et mobiles (des êtres humains en général).

Nous commençons notre étude par la mesure de taux d'occupation de la mémoire des nœuds RCSF par le firmware programmé. Le programme est stocké dans la mémoire flash, et les données dans la mémoire RAM. Le tableau 8 présente une comparaison des deux implémentations : le standard IEEE 802.15.4 et le protocole FF-MAC. Les valeurs présentées dans ce tableau sont calculées à l'aide de l'outil « avr-size » qui fait partie de « avr-gnu-toolchain ». Le tableau distingue les différents rôles des nœuds RCSF (c.-à-d. coordinateur et end-device). Nous constatons tout d'abord que le programme du coordinateur occupe plus de mémoire que celui des autres nœuds du réseau. Cela vient de l'ensemble des mécanismes additionnels dans ce nœud (FFD) par rapport aux fonctionnalités réduites dans les nœuds terminaux (RFD). Les résultats montrent aussi que les mécanismes additionnels du protocole FF-MAC consomment peu de mémoire. En effet, l'implémentation de ce protocole augmente l'occupation de la mémoire de 0.8 % pour le programme et 1.1 % pour les données dans le cas du coordinateur. Dans les nœuds RFD, ces mécanismes introduisent une augmentation de 1.1 % pour le programme et 0.3 % pour les données.

Device role	Memory usage for the standard	Memory usage for FF-MAC	Memory occupied by the extra FF-MAC features
Coordinator	Device: atmega128rfal Program : 38142 octets (29.1% Full) (.text + .data + .bootloader) Data: 2660 octets (16.2% Full) (.data + .bss + .noinit)	Device: atmega128rfal Program : 39246 octets (29.9% Full) (.text + .data + .bootloader) Data: 2841 octets (17.3% Full) (.data + .bss + .noinit)	Program: 1104 octets Data: 181 octets
End-Device	Device: atmega128rfal Program : 29366 octets (22.4% Full) (.text + .data + .bootloader) Data: 1071 octets (6.5% Full) (.data + .bss + .noinit)	Device: atmega128rfal Program : 30756 octets (23.5% Full) (.text + .data + .bootloader) Data: 1115 octets (6.8% Full) (.data + .bss + .noinit)	Program: 1390 octets Data: 44 octets

Tableau 8 : Utilisation de mémoire par l'implémentation du protocole FF-MAC et celle du standard IEEE 802.15.4

2.5.3. Validation de la nouvelle structure de la supertrame.

Dans cette partie nous présenterons les résultats obtenus sur les équipements de mesure afin de valider le nouveau format de la supertrame. Pour cela, nous avons utilisé deux méthodes : la première est la visualisation des signaux transmis dans la supertrame sur un oscilloscope afin de montrer la succession et le respect des intervalles de temps définis par le protocole. La deuxième se base sur un sniffer qui collecte toutes les trames transmises sur les réseaux et les décode pour pouvoir regarder leurs contenus.

Oscilloscope :

La figure 29 montre le format de la nouvelle supertrame capturée par un oscilloscope. Elle montre le séquençement des paquets envoyés en commençant par le Beacon, puis les transmissions dans la période CFP qui sera suivie par la transmission de la trame PRTPA. Ensuite, la période D-CFP commence avec les dimensions définies dans la trame PRTPA. Enfin, la période CAP commence juste après la fin de la période D-CFP pour donner l'accès aux nœuds non prioritaires.

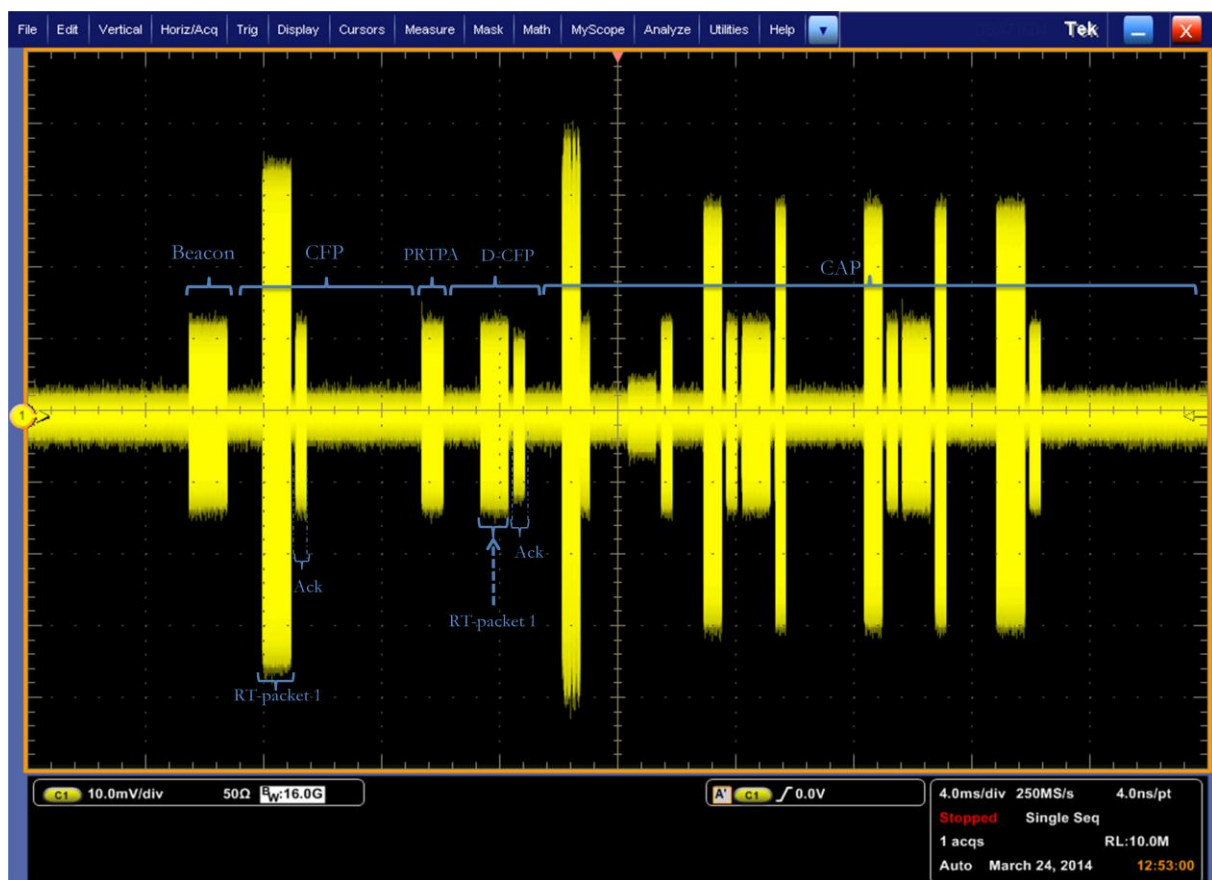


Figure 29 : Transmission des trames selon le protocole FF-MAC

L'interprétation des différents signaux est faite en comparant ces résultats avec ceux obtenus par le sniffer et qui sont présentés dans la figure 30.

Nous avons mentionné dans le deuxième chapitre que la réservation des GTS par multiples d'un slot temporel n'est pas efficace et peut entraîner un gaspillage de bande passante. La figure 29 confirme ce constat, car la période CFP dans notre exemple ne contient qu'un seul GTS d'une durée d'un seul slot. Pourtant, nous pouvons constater que le nœud propriétaire du GTS n'en a utilisé qu'une partie (environ 25 %). Par contre, le coordinateur, comme il est prévu, utilise la majorité du D-CFP pour relayer les données temps-réel à leurs destinations respectives. En effet, lors de l'allocation des D-GTS, le coordinateur calcule le temps nécessaire pour envoyer tous les paquets temps-réel en attente selon l'équation (VII). Il utilise comme unité le nombre de backoff au lieu du nombre de slots.

Analyseur de paquets :

Dans le dernier cas nous avons utilisé un sniffer faisant partie des fonctionnalités du logiciel JsensorMonitor que nous avons développé. Ce Sniffer permet de récupérer via le port série les trames reçues par un nœud en mode « promiscuous¹⁴ » et les sauvegarde dans un fichier au format PCAP. La figure 30 montre l'exemple d'un réseau de 7 nœuds, dont un coordinateur (0x0000), un nœud propriétaire d'un GTS (0x0001), un nœud destinataire des données temps-réel et propriétaire d'un D-GTS (0x0002), en plus de 4 nœuds non prioritaires. La trame diffusée par le coordinateur et marquée comme « *Unkown Command* » représente la trame PRTPA dont l'identifiant de commande MAC est « 0x0a » (non reconnu par le logiciel Wireshark).

¹⁴ Le mode promiscues « promiscuous » est un mode spécial défini dans le standard IEEE 802.15.4. Il autorise à la couche MAC du nœud configuré ainsi d'accepter toutes les trames reçues par la couche physique et les relayer à la couche supérieure. Ce mode peut être utilisé pour des raisons de supervision.

No.	Source	Source PAN	Destination	Destination PAN	Protocol	Length	Info	
16744	0x0000	0xcafe			IEEE 802.15.4	36	Beacon, Src: 0x0000	BEACON
16745	0x0001		0x0000	0xcafe	IEEE 802.15.4	24	Data, Dst: 0x0000, Src: 0x0001	
16746					IEEE 802.15.4	5	Ack	
16747	0x0000		Broadcast	0xcafe	IEEE 802.15.4	17	Unknown Command	CFP/GTS
16748	0x0000		0x0002	0xcafe	IEEE 802.15.4	24	Data, Dst: 0x0002, Src: 0x0000	
16749					IEEE 802.15.4	5	Ack	
16750	0x0005		0x0000	0xcafe	IEEE 802.15.4	25	Data, Dst: 0x0000, Src: 0x0005	PRTPA D-CFP/ D-GTS
16751					IEEE 802.15.4	5	Ack	
16752	0x0004		0x0000	0xcafe	IEEE 802.15.4	24	Data, Dst: 0x0000, Src: 0x0004	
16753					IEEE 802.15.4	5	Ack	
16754	0x0002		0x0000	0xcafe	IEEE 802.15.4	24	Data, Dst: 0x0000, Src: 0x0002	
16755					IEEE 802.15.4	5	Ack	
16756	0x0001		0x0000	0xcafe	IEEE 802.15.4	12	Data Request	
16757					IEEE 802.15.4	5	Ack	
16758	0x0000		0x0001	0xcafe	IEEE 802.15.4	24	Data, Dst: 0x0001, Src: 0x0000	
16759					IEEE 802.15.4	5	Ack	CAP
16760	0x0001		0x0000	0xcafe	IEEE 802.15.4	12	Data Request	
16761					IEEE 802.15.4	5	Ack	
16762	0x0000		0x0001	0xcafe	IEEE 802.15.4	24	Data, Dst: 0x0001, Src: 0x0000	
16763					IEEE 802.15.4	5	Ack	
16764	0x0003		0x0000	0xcafe	IEEE 802.15.4	12	Data Request	
16765					IEEE 802.15.4	5	Ack	
16766	0x0000		0x0003	0xcafe	IEEE 802.15.4	24	Data, Dst: 0x0003, Src: 0x0000	
16767					IEEE 802.15.4	5	Ack	
16768	0x0000	0xcafe			IEEE 802.15.4	34	Beacon, Src: 0x0000	

Frame 16747: 17 bytes on wire (136 bits), 17 bytes captured (136 bits)

IEEE 802.15.4 Command, Dst: Broadcast, Src: 0x0000

- Frame Control Field: Command (0x88c3)
 - Sequence Number: 6
 - Destination PAN: 0xcafe
 - Destination: 0xffff
 - Source: 0x0000
 - Command Identifier: Unknown (0x0a)
- Data (5 bytes)

Figure 30 : Partie des échanges enregistrés par notre « sniffer » et affichée sur le logiciel « Wireshark »

2.5.4. Influence du changement des périodes sur le délai (nœuds vers le coordinateur)

Dans ce scénario, nous allons illustrer la première limitation que nous avons identifiée dans le standard IEEE 802.15.4, puis comparer les résultats obtenus avec ceux de notre protocole FF-MAC et le résultat théorique discuté dans la section 2.3 (chapitre II).

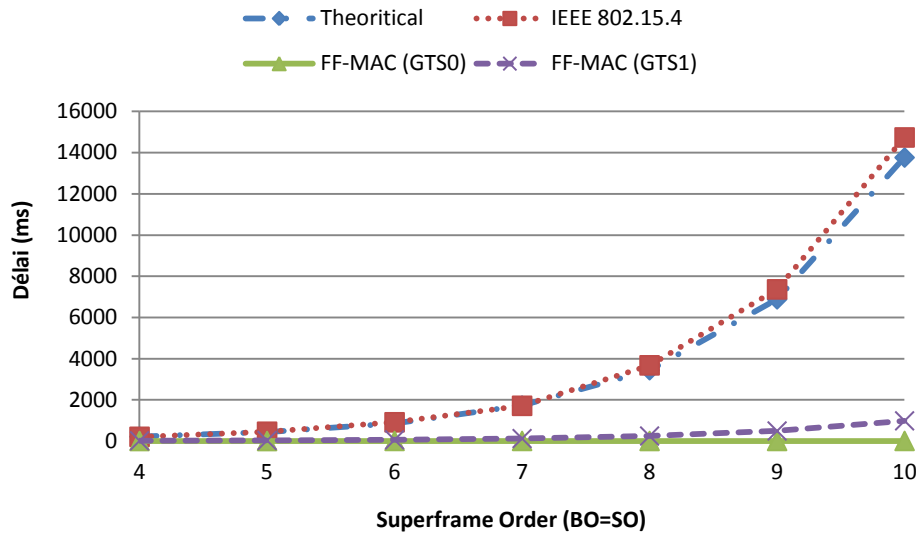


Figure 31 : Délai de communication entre le coordinateur et un nœud du réseau propriétaire d'un GTS

Comme nous pouvons le constater, le positionnement de la période CFP à la fin de la supertrame entraîne un délai supplémentaire. Ce délai devient plus important lorsque les slots de la supertrame se dilatent par de l'augmentation du paramètre SO. Nous constatons aussi que le délai est proche du délai théorique, car nous avons reproduit les mêmes conditions présentées dans le chapitre II (un seul GTS composé d'un slot). Cependant, le changement introduit par notre approche a résolu ce problème en réduisant le délai d'envoi vers et depuis le coordinateur dans les GTS (environ 7 ms). En effet, les données sont transmises immédiatement après la réception du Beacon. Pour le premier GTS (GTS 0), ce délai reste insensible au changement de la durée des slots. Les résultats selon le protocole FF-MAC sont identiques pour toutes les valeurs de SO et ne dépendent plus de la durée de la période active, car les paquets sont transmis directement après la réception du Beacon. Pour le deuxième GTS (GTS 1), le délai augmente faiblement du fait que le nœud doit attendre l'arrivée de son GTS pour pouvoir transmettre ses paquets. Ce temps dépendra de la valeur du paramètre SO et, par conséquent, de la durée des slots. Cependant, la valeur du délai reste faible, car la CFP est placée au début de la supertrame. Ceci est illustré à la figure 31 où la courbe du délai de bout en bout dans le deuxième GTS est légèrement au-dessus de celle du premier GTS, mais qui reste nettement inférieure à celle du standard.

2.5.5. Le relais rapide dans la même supertrame

Parmi les améliorations apportées par le protocole FF-MAC, il y a la possibilité de retransmettre des données temps réel dans la même supertrame. Nous avons montré par

simulation que l'approche proposée introduit des améliorations importantes par la réduction du délai de communication pour les scénarios multi-sources multi-sink. Ces améliorations sont confirmées à l'aide du prototypage. Comme nous l'avons montré dans le chapitre II, le délai de bout en bout dans ce genre de scénarios est influencé par les deux paramètres SO et BO. Le premier détermine la durée de la période active et des slots, tandis que le deuxième définit la durée totale de la supertrame (la période de sommeil incluse). Les résultats suivants ont été mesurés en fonction de ces deux paramètres.

La figure 32 montre le problème d'envoi sur plusieurs superframes en utilisant la méthode d'accès CSMA-CA comparée aux améliorations apportées par le protocole FF-MAC. Nous constatons que pour la méthode d'accès CSMA-CA, le délai est proportionnel à la durée de la supertrame. La majorité des paquets arrivent dans la supertrame qui suit celle de l'envoi initial. D'autres arrivent après deux supertrames et encore moins arrivent dans les autres supertrames qui suivent. Dans cet exemple, les paramètres de la supertrame ont été choisis pour avoir un rapport cyclique de 12.5% (BO = 7 et SO = 4). On peut constater que les délais obtenus sont bien des multiples de 2 secondes, ce qui correspond à la durée totale de la supertrame (Intervalle du Beacon) :

$$\begin{aligned}\text{Durée de la supertrame} &= aBaseSuperframeDuration \times 2^{BO} \\ &= 122880 \text{ symboles} = 1966,08 \text{ ms}\end{aligned}$$

Le protocole FF-MAC quant à lui permet de délivrer les paquets à leurs destinations rapidement au début de la supertrame. Le délai moyen pour le protocole FF-MAC est de 32,659 ms contre un délai moyen de 2287,945 ms pour la méthode CSMA-CA

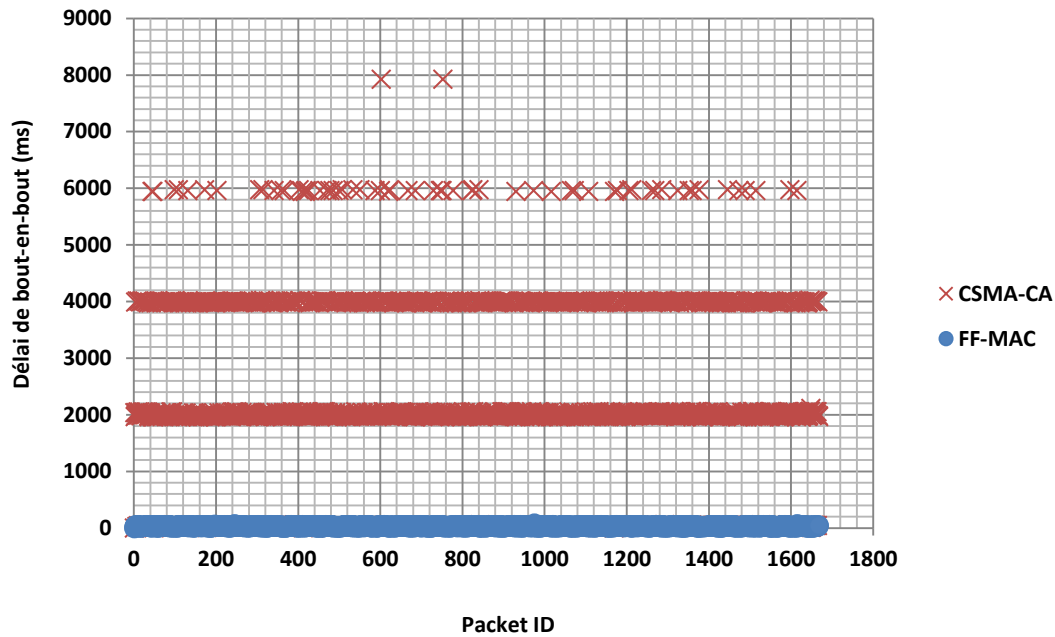


Figure 32 : Délai de bout en bout pour chaque paquet (BO=7, SO=4)

Dans la figure 33, nous essayons de mettre en évidence le problème identifié dans le deuxième chapitre. Nous avons évalué l'influence du changement de la durée de supertrame sur le délai de communication. Le rapport cyclique est de 100 %. Donc, la période active occupe la totalité de la supertrame. Nous constatons que l'augmentation de la période active (autrement dit, l'augmentation de SO) affecte clairement le délai de communication. Le délai enregistré est très proche de la durée de la supertrame, car le paquet est récupéré par le destinataire au début de la supertrame suivante. Ce délai peut dépasser considérablement la durée de la supertrame en présence des collisions. Dans ces scénarios, la réception des paquets au début de la supertrame est très probable grâce à l'absence des collisions. Néanmoins, le délai mesuré en utilisant le protocole FF-MAC est très faible pour toutes les valeurs de SO. Ceci vient du fait que les paquets sont transmis et relayés au début d'une même supertrame. Ce délai est tout de même légèrement affecté par ce paramètre, car la durée des slots dépend elle aussi de du paramètre SO.

Le capteur réel qui joue le rôle de coordinateur du réseau doit limiter la taille de la file d'attente des transmissions indirectes. En effet, contrairement à la simulation, la taille de la mémoire d'un nœud capteur réel est très limitée (c.-à-d. 16384 octets pour le coordinateur, en l'occurrence). Ainsi, le nombre de paquets reçus par ce dernier peut être plus que sa capacité de stockage. Dans ce cas, le coordinateur appliquera la politique d'élimination des paquets en excès « tail drop ». Ainsi, en vue de protéger les paquets prioritaires, nous avons proposé l'utilisation d'une file d'attente virtuelle (Virtual Queue « VQ ») pour les paquets temps-réel. En effet, cette

file d'attente est dérivée de la file d'attente principale qui est utilisée pour tout le reste des trafics. Le coordinateur réserve un nombre de tampons pour les paquets prioritaires pour cette file d'attente virtuelle, en se basant sur la quantité du trafic prévue par supertrame, c'est-à-dire le nombre de GTS de réception réservés et le taux de transmission des trames.

Par exemple, s'il y a un seul GTS en mode réception et la fréquence d'échantillonnage est égale à l'intervalle du beacon « BI », la file d'attente virtuelle ne contiendra qu'un seul tampon de paquet. Ainsi, même avec une taille aussi petite de la file d'attente virtuelle, le protocole FF-MAC peut protéger ces paquets prioritaires, car le coordinateur libère cette file d'attente très rapidement avant la réception d'autres paquets temps-réel. La taille peut changer dynamiquement en fonction de la situation du réseau.

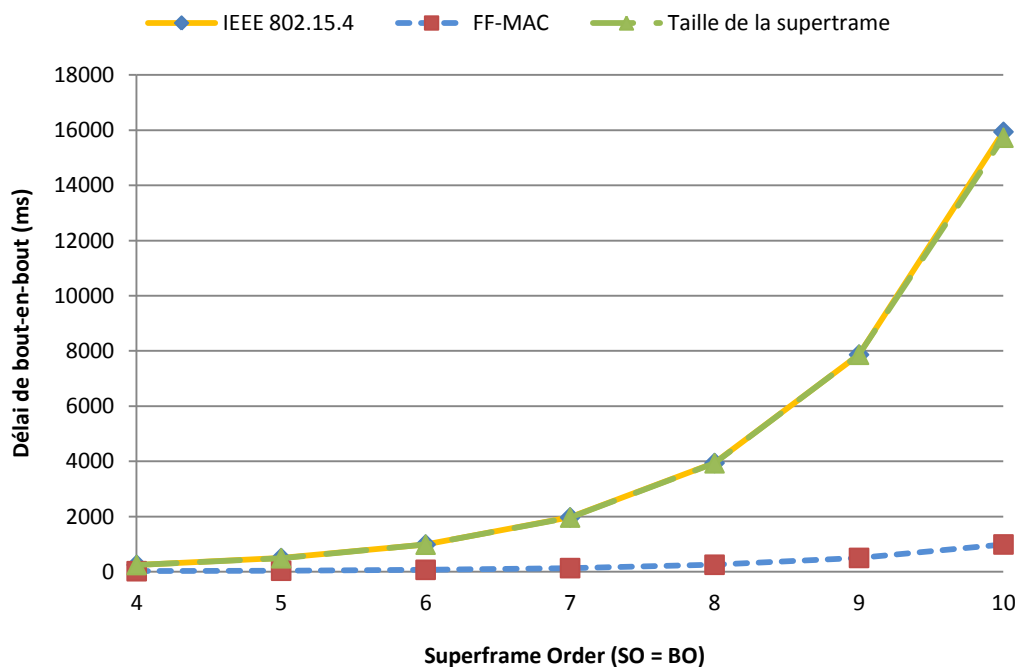


Figure 33 : Délai de bout en bout en fonction du paramètre SO (BO = SO)

La figure 34 montre l'influence de la période de sommeil¹⁵ sur les performances en termes de délai pour les deux protocoles. Le changement de BO seul permet de modifier la période de sommeil tout en fixant la durée de la période active et celle des slots. On constate que les valeurs des délais obtenus pour le protocole FF-MAC sont faibles et ne varient pas en fonction du paramètre BO. Ceci est justifié par le fait que les trames ne sont pas sauvegardées dans le coordinateur durant la période de sommeil, mais ils sont plutôt retransmis au début de la

¹⁵ Le paramètre SO est fixé à 4

supertrame. Par contre, le délai du standard est sensible à l'augmentation du temps de sommeil ce qui engendre des délais supplémentaires.

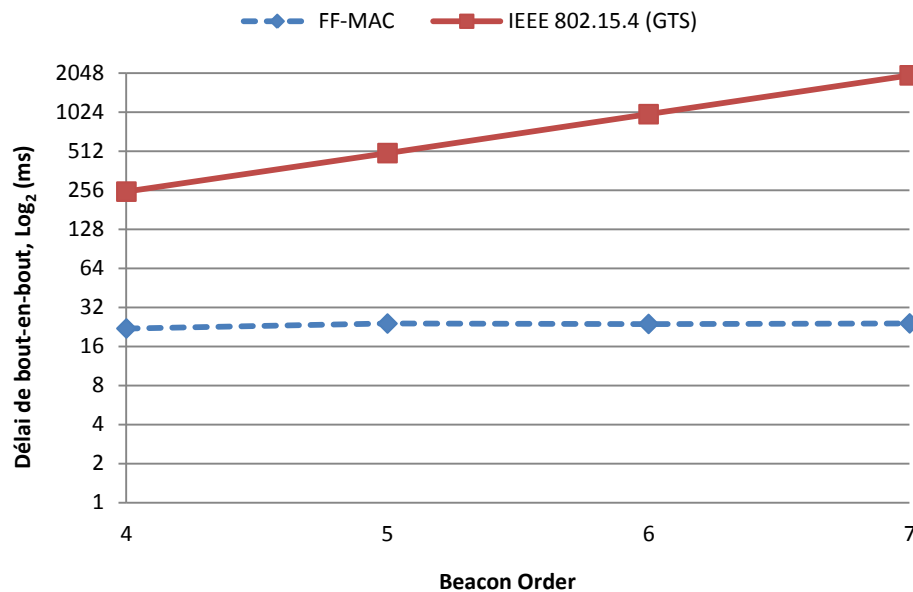


Figure 34 : Délai de bout-en-bout en fonction du paramètre BO (SO=4)

Les résultats peuvent être analysés par un autre paramètre qui englobe les valeurs de BO et SO. Il s'agit du rapport cyclique. Ce dernier reflète le pourcentage d'activité du réseau en entier par rapport au temps de silence.

Remarque : l'activité des nœuds peut être très inférieure au rapport cyclique défini par les paramètres BO et SO, notamment lorsque l'attribut MAC `macRxOnWhenIdle` prend la valeur « FAUX ». En effet, les deux paramètres BO et SO contrôlent la durée de la supertrame et celle de la période active respectivement. Néanmoins, lorsque la valeur de l'attribut `macRxOnWhenIdle` est nulle, le nœud peut se mettre en mode de sommeil lorsqu'il n'a des données ni pour transmettre ni pour recevoir. Cela peut se faire dans la période active de la supertrame.

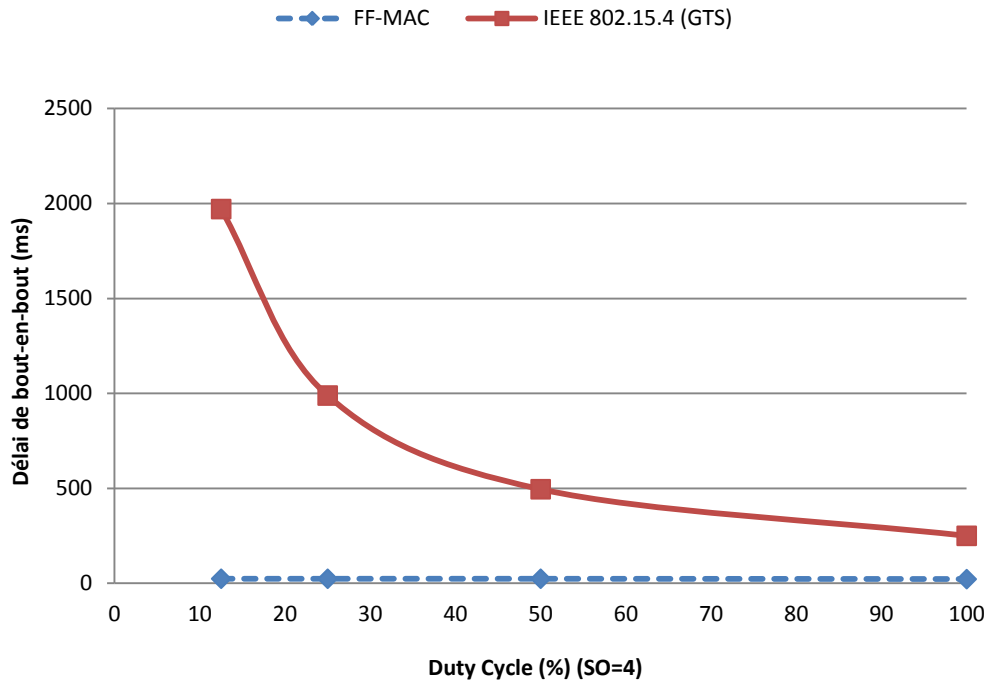


Figure 35 : Délai de bout-en-bout en fonction du rapport cyclique (SO=4)

Nous constatons également que le délai mesuré pour le mécanisme de GTS du standard est sensible au rapport cyclique, tandis que le protocole FF-MAC offre un délai faible et invariant.

2.5.6. Aspect déterministe du protocole FF-MAC

Nous constatons dans la figure 36 que la variation du nombre de nœuds dans le réseau influence sur le délai de bout en bout dans le cas du standard. En effet, ce problème est hérité de l'algorithme CSMA-CA utilisé dans la réception, qui est très sensible au passage à l'échelle. Par contre, de par son caractère temps réel, le protocole FF-MAC permet aux nœuds propriétaires des GTSs d'avoir un délai de bout en bout qui ne dépend pas du nombre de nœuds. Toutes les transmissions se font sans contentions de bout en bout grâce à l'utilisation des périodes CFP et D-CFP. Cependant, si l'on souhaite assurer une communication déterministe pour tous les nœuds du réseau, le nombre de ces derniers sera limité pour chaque cluster en fonction de plusieurs paramètres, à savoir : le paramètre SO, la taille des données à transmettre et le type de la couche physique utilisé. Par exemple, si les nœuds utilisent la couche PHY 2.4 GHz, avec un ordre de supertrame SO égal à 4 et un MSDU de 30 octets, le réseau peut supporter jusqu'à 26 nœuds prioritaires (13 GTS et 13 D-GTS). Pour des trames d'un MSDU de 118 octets¹⁶ le nombre de

¹⁶ C.-à-d. la taille maximale autorisée par la couche physique lorsque la couche MAC introduit l'entête la plus courte : PSDU de 127 octets),

nœuds se réduit à 22 nœuds (11 GTS et 11 D-GTS). En effet, la taille des D-GTS est inférieure à celle des GTS, car le D-GTS donne seulement la durée nécessaire pour transmettre la quantité de données souhaitée. Les GTS quant à eux, sont réservés en multiple de slots. Notons que dans ce calcul, nous avons ignoré que la contrainte qui limite le nombre de GTS à 7.

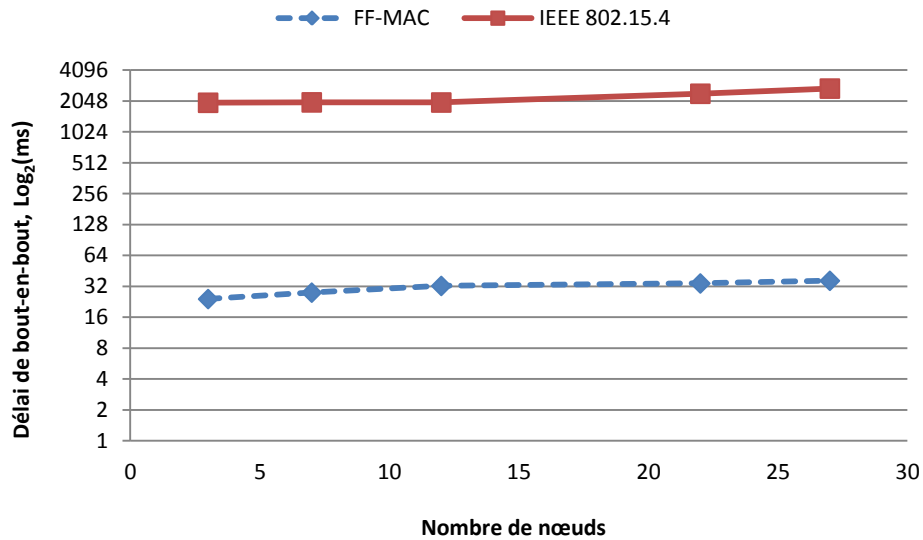


Figure 36 : Délai de bout-en-bout en fonction de la densité des nœuds

2.5.7. Impact du protocole FF-MAC sur les performances de l'algorithme CSMA-CA

Le protocole FF-MAC utilise la période dynamique D-CFP afin de retransmettre les données temps-réel en attente. Cette période est créée à partir de la période CAP. Ainsi, lorsque la période D-CFP existe, la période CAP devient plus petite et par conséquent, les nœuds auront moins d'espace pour transmettre leurs paquets. La Figure 37 montre cet impact sur le délai mesuré pour les paquets non prioritaires dans les deux protocoles, les performances du standard sont meilleures, mais reste très proches de celles du protocole FF-MAC. Cela vient du fait que ce dernier utilise des D-GTS d'une durée efficace qui dépend de la quantité des données disponibles et non pas d'une durée prédéfinie (comme les GTS).

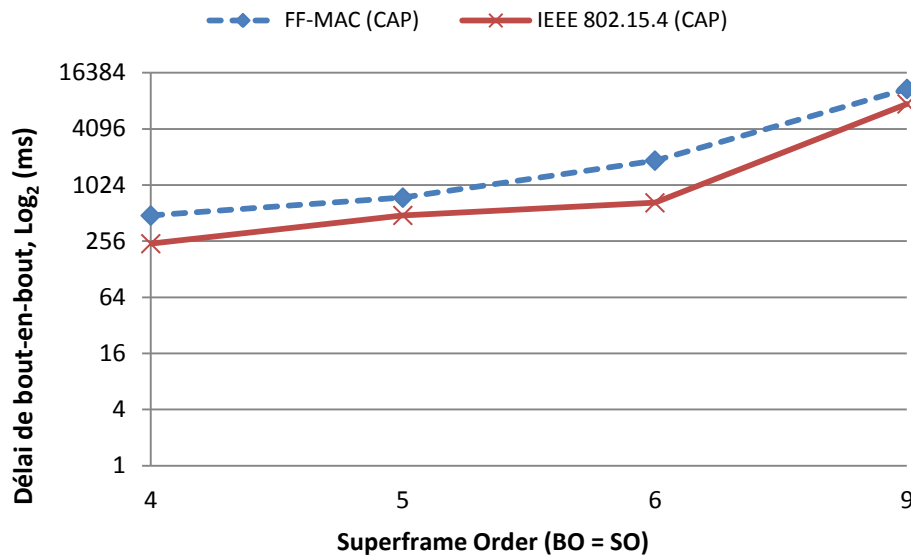


Figure 37 : Comparaison des performances des nœuds utilisant la période CAP (CSMA-CA) dans le cas du protocole FF-MAC et le standard IEEE 802.15.4 (DC = 100%)

L'utilisation de la bande passante pour les nœuds ayant des données¹⁷ temps-réel est présentée dans la figure 38. Premièrement, le protocole FF-MAC utilise la bande passante de manière plus efficace que le standard. Cela vient de la capacité du premier de garantir la délivrance des données prioritaires dans la liaison montante « uplink » (nœuds sources vers le coordinateur) et la liaison descendante « downlink » (coordinateur vers les nœuds destinations). Le standard ne permet de garantir que la communication dans le sens montant. Notons que les erreurs de transmission sont corrigées grâce aux retransmissions au niveau de la sous-couche MAC. Deuxièmement, on remarque une stabilité dans les valeurs des débits mesurées dans le cas de notre approche contre un impact négatif du nombre de nœuds sur le débit dans le cas du standard. On peut justifier cela par l'utilisation de l'algorithme CSMA-CA par ce dernier dans le sens descendant de la communication, contre l'utilisation des D-GTS déterministe dans notre protocole.

¹⁷ Les nœuds envoient une trame chaque BI (intervalle de Beacon).

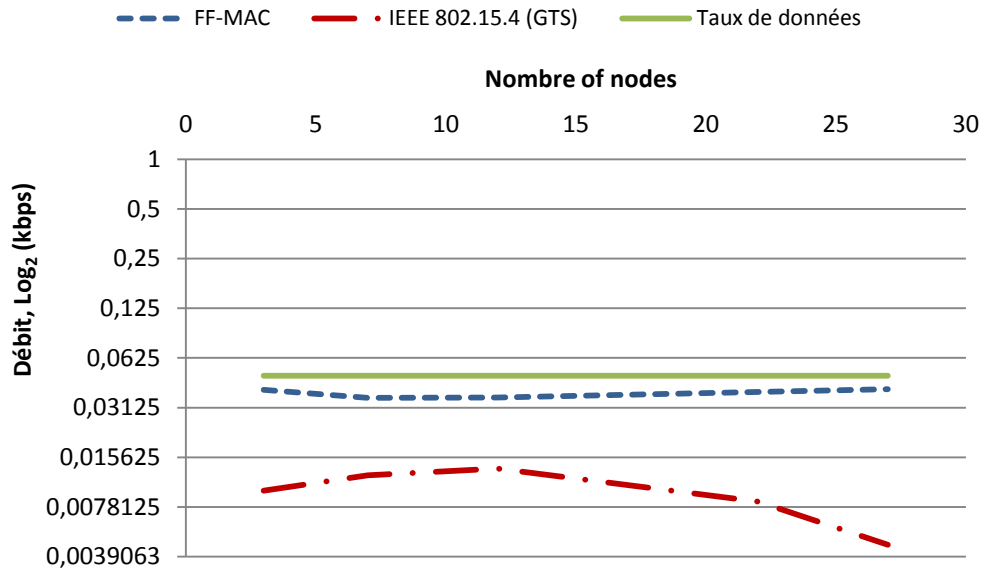


Figure 38 : Débits mesurés en fonction des nombres de nœuds (DC=12.5%)

2.6. Remarque importante

Dans la conception et l'implémentation du protocole MAC, nous avons essayé de garder un minimum de compatibilité avec le standard IEEE 802.15.4. En effet, il est possible d'utiliser le même code du protocole FF-MAC dans un réseau IEEE 802.15.4 lorsque l'utilisation des GTS est désactivée. Cette compatibilité a été vérifiée et validée sur les nœuds RCSF réels.

3. Résumé des améliorations

L'amélioration du standard IEEE 802.15.4 s'est faite sur plusieurs étapes. Tout d'abord, nous avons inversé l'ordre des deux périodes CAP et CFP en mettant la période CFP au début de la supertrame. Ce nouveau format donne la possibilité au coordinateur de retransmettre les données temps-réel dans la même supertrame. Ensuite, nous avons introduit un nouveau type de trames MAC appelé PRTPA permettant de donner les informations sur les paquets temps-réel transmis récemment dans la CFP (placé au début de la supertrame). Enfin, nous avons introduit une nouvelle période dynamique appelée D-CFP basée sur l'ordonnancement annoncé dans la trame PRTPA désignant les nœuds RCSF récepteurs ayant des données temps-réel en attente. Nous avons gardé la trame beacon intacte afin de garder un minimum de compatibilité avec le standard original, seule la signification d'un seul champ a été changée.

4. Conclusion du chapitre

Les résultats obtenus présentent les améliorations prévues par notre approche et montrent que le protocole FF-MAC est plus performant que le mécanisme des GTS du standard IEEE 802.15.4 dans tous les scénarios étudiés. Le protocole FF-MAC permet d'atteindre des délais très faibles. Nous avons montré qu'en utilisant notre protocole, le paramètre BO et la densité des nœuds n'ont aucun impact sur le délai pour les nœuds qui disposent de GTS/D-GTS, grâce à un acheminement déterministe de bout en bout. Cependant, le nombre de nœuds qui peuvent profiter d'un accès déterministe de bout en bout est limité par la capacité des GTS et D-GTS dans la supertrame. Cette dernière est affectée par plusieurs facteurs comme : la limitation des GTS à un maximum de 7 GTS par supertrame, la taille de la période CAP doit être égale à $aMinCAPLength$ au minimum. Le changement de la durée des slots a une influence faible sur le délai mesuré (ceci dépend de la taille des données). Ceci montre que le protocole FF-MAC est adapté aux applications dans lesquelles circule un trafic avec des priorités de données et des besoins de qualité de service hétérogènes.

L'évaluation des performances a été effectuée par simulation et par prototypage ce qui nous a permis de prouver les améliorations apportées aussi bien sur le plan théorique que pratique.

CHAPITRE IV :

Implémentation du Standard IEEE 802.15.4 sous le Simulateur NS-3

La simulation joue un rôle très important dans la recherche scientifique. Elle permet, d'une part, de valider les protocoles proposés d'un point de vue théorique et, d'autre part, de dépasser les limites imposées par les contraintes expérimentales comme le passage à l'échelle. Dernièrement, les chercheurs commencent à étudier la possibilité de faire une intégration de la simulation dans les réseaux réels. Un exemple de cette nouvelle utilisation se manifeste dans les systèmes CLAS (Closed-Loop Adaptive System) dans lesquels le simulateur est utilisé comme un outil qui veille sur la stabilité du réseau en prévoyant son comportement sur la base des données recueillies du réseau réel. La simulation peut aussi exécuter des algorithmes à moindre coût et envoyer les résultats au système réel pour éviter toute charge du réseau et dépense des ressources limitées (p. ex. consommation des cycles de processeur à cause du traitement des messages de gestion, occupation de la bande passante par les échanges des messages de mises à jour ...). Ceci représente un changement de paradigme vis-à-vis du rôle de la simulation dans le domaine des réseaux en général, et celui des RCSF en particulier.

Dans ce chapitre nous allons discuter tout d'abord les motivations pour développer un modèle WPAN modélisant le standard IEEE 802.15.4 sous le nouveau simulateur NS-3, avec un aperçu sur les modèles existants. Ensuite, nous

présenterons l'architecture de ce modèle et divulguons les fonctionnalités assurées par ses composantes. Enfin, nous analyserons les résultats de l'évaluation de performances du modèle proposé.

1. Modèles existants

À partir de l'année 2003, après la publication de la première version du standard IEEE 802.15.4 (c.-à-d. IEEE 802.15.4-2003), un ensemble de modèles de simulation qui modélisent ce dernier ont été proposés pour différentes plateformes de simulation. L'objectif est de proposer des outils permettant d'étudier ce standard et d'évaluer ses performances et ceux de ses améliorations. Parmi les simulateurs des réseaux les plus connus qui proposent des modèles pour le standard IEEE 802.15.4, on trouve : NS-2, OMNeT++ [114], OPNET [115]...

Le simulateur NS-2 était l'un des premiers qui ont fourni un modèle pour le standard IEEE 802.15.4. Ce modèle, proposé dans [106], supporte la plupart des fonctionnalités du standard IEEE 802.15.4. Par contre, les mécanismes de sécurité et de gestion des GTS ne sont pas disponibles jusqu'à présent (version 2.35). Un autre modèle a été proposé dans [116] pour le simulateur OPNET. Il est destiné principalement à l'évaluation des performances du mécanisme des GTS. Ainsi, il supporte seulement le mode avec Beacon « Beacon-enabled » et le mécanisme de gestion des GTS. Le mode sans beacon, y compris l'algorithme CSMA-CA non slotté, n'est pas implémenté. Aussi, la gestion du PAN n'est pas supportée. Pour le simulateur OMNeT++, les packages Castalia [117] et MiXiM [118] proposent aussi des modèles de simulation complets, publiés dans [119], et qui disposent de plusieurs fonctionnalités.

2. Motivations

Vu l'intérêt du standard IEEE 802.15.4 dans le domaine des réseaux de capteurs sans fil, nous avons développé un nouveau modèle pour le simulateur NS-3 basé essentiellement sur le modèle WPAN du simulateur NS-2. En effet, ce dernier est l'un des modèles les plus matures et les plus utilisés par la communauté des chercheurs dans le domaine des RCSF. Cela permettra de réduire le temps d'apprentissage pour ce modèle et de faciliter la migration de leurs approches qui sont basées sur ce standard. La motivation vient aussi du fait que le nombre d'utilisateurs du simulateur NS-3 devient de plus en plus large et il y a sûrement un besoin d'une implémentation qui modélise les couches basses pour les RCSF. De plus, ce travail s'inscrit dans le cadre d'un projet dans notre équipe de recherche dans lequel nous proposerons une pile protocolaire complète pour les technologies : Internet des Objets (IoT) et Web des Objets (WoT). Ces deux technologies qui constituent la l'un des tendances futures des RCSF ont adopté le standard IEEE 802.15.4 pour les couches basses, d'où l'intérêt de notre modèle dans ce domaine.

3. Aperçus du Simulateur NS-3

NS-3 ou « Network Simulator version 3 » est un nouveau simulateur réseau dont le développement a commencé en 2006 comme étant le successeur de l'ancien simulateur NS-2 « Network Simulator version 2 ». En effet, ce dernier souffre de plusieurs limitations et problèmes techniques. De plus, un ensemble de modèles de protocoles ne sont pas très précis surtout ceux liés à la communication sans fil qui ne faisaient pas partie du projet de départ, mais qui étaient intégrés ultérieurement dans le simulateur. Ceci a donné une motivation aux chercheurs pour développer une nouvelle version qui remplacera l'ancienne plateforme de simulation et combler ses lacunes.

Néanmoins, NS-3 n'est pas une nouvelle version de NS-2, mais un simulateur totalement différent de ce dernier qui contient aussi bien de nouveaux modèles que d'anciens modèles matures de NS-2 et des autres simulateurs. NS-3 est développé pour les systèmes internet sous la licence GNU GPLv2. Il implémente de nouveaux concepts qui tirent profit des dernières avancées dans le domaine du développement informatique, surtout celles liées au langage C++. Les concepts clé de son succès sont la modularité, la réutilisabilité et la facilité d'extension.

NS-3 fait partie de la famille des simulateurs réseau à événements discrets « discrete-event ». Il est écrit entièrement en C++. Ainsi, le développeur peut profiter des multiples mécanismes utiles de ce langage comme, à titre d'exemple : l'utilisation des « Template », les pointeurs intelligents et les modèles de conception « design patterns ». Ce simulateur se distingue des autres simulateurs par l'ensemble des mécanismes de haut niveau comme : les événements et les connexions basés sur les « callbacks » (Callback-driven events and connections) et un noyau flexible avec l'utilisation des assistants. Ce simulateur, met aussi l'accent sur l'émulation en s'alignant avec les interfaces utilisées par les systèmes réels, comme : les sockets, les « Network Device » et la création de paquets sous forme de flux binaires (serialisation/deserialisation), sans oublier ses multiples options de traçage qui donnent accès à des sorties de résultats sous plusieurs formats.

Une étude comparative a été réalisée dans [120]. Elle a pour but d'évaluer les performances d'un ensemble de simulateurs en terme de vitesse d'exécution, d'occupation de la mémoire et de temps de calcul. Les résultats obtenus ont montré que NS-3 offre globalement les meilleures performances de simulation parmi les simulateurs étudiés.

4. Conception et implémentation

4.1. Architecture globale

Notre modèle WPAN adopte la même architecture d'un nœud définie dans le standard IEEE 802.15.4. Cette architecture est illustrée dans la figure 4 (chapitre I) et la figure 39.

Les couches physiques (PHY), MAC et SSCS en plus des couches supérieures communiquent entre elles de manière exclusive via des interfaces standard à l'aide des « Callbacks » de NS-3. Cette nouvelle conception permet une meilleure séparation des couches afin de respecter le concept d'abstraction des couches définie dans le modèle OSI et de permettre une flexibilité d'extension ou d'intégration des différentes couches citées ci-dessus dans n'importe quelle autre pile protocolaire. À côté de cette architecture, nous avons adopté la notion de carte réseau (Network Device). Le concept des « Network Device » qui dérive des systèmes d'exploitation réels (Linux) est très utilisé dans le simulateur NS-3. Ce mécanisme permet d'homogénéiser les différentes couches de liaison de données pour qu'elles soient accessibles par les couches supérieures à travers des interfaces similaires. Par conséquent, il offre un certain niveau d'abstraction pour les couches supérieures. En l'occurrence, aucune modification n'est nécessaire dans les couches supérieures pour supporter ce nouveau modèle.

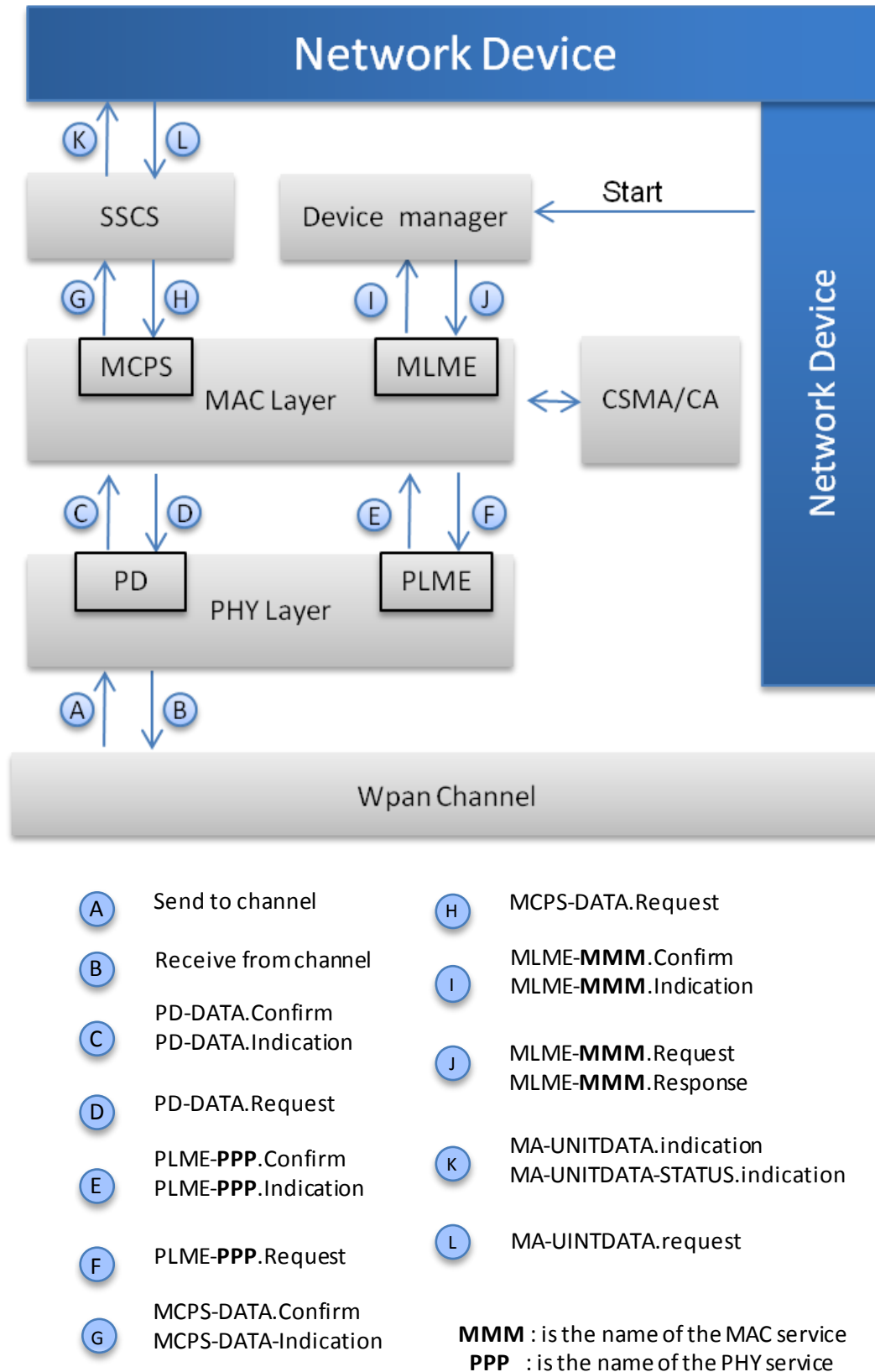


Figure 39 : Architecture du modèle WPAN

4.2. Composantes du modèle WPAN :

Le modèle WPAN est constitué d'un ensemble de composantes qui modélisent les différentes couches et les mécanismes du standard IEEE 802.15.4.

4.2.1. Couche physique : WpanPhy

Dans ce modèle, toutes les couches physiques définies dans le standard IEEE 802.15.4-2006, y compris les versions optionnelles, sont supportées. En effet, les différents débits binaires, débits de symboles, les canaux et les plages de fréquence sont disponibles afin de donner à l'utilisateur la possibilité d'évaluer les protocoles avec les différentes couches physiques proposées par le standard. Le module WpanPhy est responsable de l'envoi et de la réception des données depuis et vers le média physique (WpanChannel dans notre cas), la détection d'énergie dans le canal « ED », la vérification de la disponibilité du canal « CCA » et la sélection de la fréquence du canal à utiliser pour la transmission des trames. Il calcule également la qualité du lien « LQI » des paquets reçus et remonte cette information à la sous-couche MAC. Tous ces services sont offerts à la sous-couche MAC à travers des interfaces standard représentées par les « Callbacks » PLME « Physical Layer Management Entity » et PD « PHY Data ». Les attributs de la couche physique sont gérés par le module Ppib qui implémente la base de données PPIB « PHY PAN Information Base ».

Remarque : dans notre modèle, les techniques de modulation ne sont pas simulées. Néanmoins, pour que deux nœuds puissent communiquer, la fréquence de transmission et les types de modulations utilisées par l'émetteur et le récepteur, à savoir : BPSK, ASK ou O-QPSK, doivent être identiques.

4.2.2. Sous-couche MAC : WpanMac

Ce module offre une implémentation des différents modes MAC définis par le standard IEEE 802.15.4, à savoir ; le mode avec beacon activé (Beacon-Enabled « BEn ») et le mode beacon non activé (Non Beacon-Enabled « NBEn »). Un troisième mode sans beacon (BeaconLess « BL ») est supporté pour les réseaux « mesh ». Ce dernier mode n'exige aucune association ou gestion centralisée du réseau. De plus, l'accès au médium est géré par la méthode d'accès CSMA-CA non slottée. De plus, les nœuds sont similaires et ne peuvent utiliser que les adresses MAC en format 64-bits. En ce qui concerne le mode NBEn, les nœuds peuvent utiliser les adresses optimisées au format 16-bits après leurs associations au coordinateur. Ce mode utilise également la version non slottée de la méthode d'accès CSMA-CA. Le mode BEn par contre, est

un mode synchronisé. Les nœuds associés au coordinateur du PAN organisent leurs envois selon le format de la supertrame dont la structure est illustrée à la figure 4. L'utilisateur peut configurer les paramètres, tels que l'ordre du beacon « BO », l'ordre de la supertrame « SO » et la capacité du nœud (FFD ou RFD).

Le module WpanMac est constitué des composantes illustrées à la figure 40. Le noyau de la couche MAC fait le lien entre les différentes composantes et offre des interfaces de communication avec les couches adjacentes (c.-à-d. PHY et SSCS). La composante CSMA-CA modélise l'algorithme CSMA-CA avec ses deux variantes (slottée et non slottée) pour l'envoi des trames de données et des commandes MAC. Elle vérifie la possibilité d'envoi des données et renvoie le résultat au noyau de la couche MAC « WpanMac (Core) » pour transmettre ou différer la transmission de la trame. Le dispatcher, quant à lui, centralise la gestion des tâches en cours d'exécution afin d'en éviter les conflits. Pour la configuration de la couche MAC, nous avons mis à disposition un gestionnaire de la base de données MPIB «MAC PAN Information Base» pour la lecture, l'écriture et la vérification de la validité des valeurs des attributs.

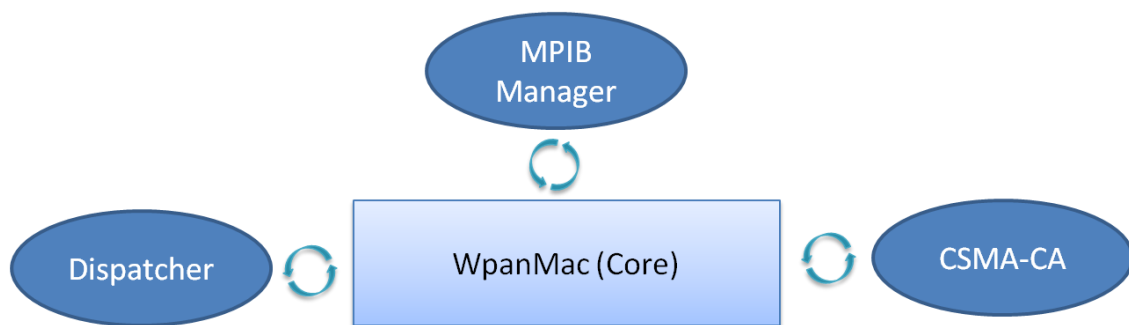


Figure 40 : Architecture interne du module WpanMac

Nous visons dans nos futurs travaux d'ajouter une nouvelle composante pour le mécanisme de gestion des GTS. Jusqu'à présent, la gestion de la commande de demande de GTS est implémentée, mais l'utilisation des slots n'est pas encore supportée.

Pour la vérification de l'intégrité des données transmises, nous avons implémenté le mécanisme FCS en se basant sur l'algorithme 16-bit ITU-T CRC « cyclic redundancy check » puisqu'il est recommandé par le standard IEEE 802.15.4. Ce CRC pourra être utilisé dans la simulation pour détecter les erreurs de transmission dans les nœuds récepteurs. Cependant, nous avons donné à l'utilisateur la possibilité de le désactiver afin d'accélérer la simulation.

4.2.3. Sous-couche de convergence : WpanSscs

SSCS « Service-Specific Convergence Sub-layer » est tout simplement une couche d'adaptation qui fournit une interface entre une instance de la sous-couche IEEE 802.2 LLC et la sous-couche MAC du standard IEEE 802.15.4 à travers les primitives MCPS « MAC Common Part Sublayer ». Dans notre modèle, la sous-couche SSCS permet d'adapter les échanges entre la sous-couche MAC et le module WpanNetDevice qui sera connecté aux couches supérieures (voir le paragraphe 4.2.5). Toutes les communications s'effectuent via les primitives suivantes¹⁸ :

- MA-UNITDATA.request
- MA-UNITDATA.indication.
- MA-UNITDATA-STATUS.indication

4.2.4. Gestionnaire du nœud : WpanDeviceManager

Le gestionnaire des nœuds WPAN est un module important pour un nœud WPAN. Ce module, qui ne fait pas partie du standard IEEE 802.15.4, est ajouté pour gérer et configurer les nœuds. En effet, les modèles des couches supérieures existants sous NS-3 n'ont aucun mécanisme qui prendra en charge ces tâches. Ainsi, ce module dispose de toutes les fonctions de l'interface de gestion de la sous-couche MAC via le point d'accès au service MLME-SAP. La connexion se fait à l'aide des « callbacks » des services suivants :

- Démarrage des nœuds.
- Le scan du médium sans fil
- Association au coordinateur
- Désassociation 'Dissociation' du coordinateur
- Synchronisation/Récupération après perte de synchronisation.
- Notification de la réception du Beacon
- Paramétrage des couches MAC et PHY (Get/Set)

¹⁸ Définis dans l'annexe "A" de la spécification du standard IEEE 802.15.4-2006 [121]

- Gestion de l'état d'orphelinat du nœud (perte de connectivité avec le coordinateur parent)
- Commande de demande du GTS
- Demande des données en attente
- Réinitialisation du nœud

Cependant, ce module pourra être ignoré lors de l'utilisation d'une couche supérieure susceptible d'utiliser l'interface de gestion comme le protocole ZigBee. Grâce au mécanisme des callbacks, aucun changement n'est requis dans la sous-couche MAC. Il suffit de connecter les « callbacks » MLME-SAP à la nouvelle couche supérieure à travers le « WpanNetDevice ».

4.2.5. WpanNetDevice

Le module « WPAN network device » (WpanNetDevice) représente simultanément le matériel simulé et le pilote logiciel. Un exemple de cette implementation est le modèle du standard IEEE 802.11 (wifi) existant. Dans notre modèle, la composante « WPAN network device » joue le rôle d'un conteneur qui englobe la pile des protocoles proposés dans le standard IEEE 802.15.4 (MAC, PHY & SSCS) en plus de la gestion du nœud (4.2.4), le cas échéant. Ce module est aussi responsable de la liaison des interfaces des couches adjacentes en utilisant le mécanisme des callbacks. Toute configuration de cette pile protocolaire est centralisée exclusivement dans ce module.

De plus, le module WpanNetDevice remplace la sous-couche 802.2 LLC en jouant le rôle d'intermédiaire entre la sous-couche SSCS et les couches supérieures via les interfaces standards mentionnées dans la figure 39 (les primitives K and L). L'utilisateur peut désactiver l'en-tête LLC/SNAP qui peut s'avérer inutile dans certains protocoles de niveau supérieur (6LoWPAN à titre d'exemple).

4.2.6. Canal de transmission sans fil

Le média sans fil réel est représenté dans le modèle WPAN par le module « WpanChannel ». Ce dernier est utilisé pour simuler la propagation des ondes radios (c.-à-d. les paquets) transmises par chaque nœud à tous les nœuds dans son rayon de couverture « POS » (Personal Operating Space). Pour que le signal soit intelligible pour ces nœuds récepteurs, ces derniers doivent écouter sur le même canal et utiliser une technique de modulation identique à

celle de l'émetteur. Un seul canal radio est utilisé pendant la simulation pour toutes les couches physiques et les fréquences choisies dans le scénario simulé.

Étant une extension de la classe « Channel » de NS-3, le module « WpanChannel » peut utiliser tous les modèles d'affaiblissement de propagation et ceux du délai de propagation existants sous NS-3. Ces modèles permettent d'estimer le délai de propagation et l'atténuation du signal qui, lorsqu'elle est couplée avec la puissance d'émission et le seuil de réception du nœud, déterminent la portée radio.

4.2.7. Modèle d'énergie

Notre modèle d'énergie est basé sur le Framework d'énergie proposé dans [122] et intégré officiellement sous NS-3. Pour simuler la batterie, le modèle « WpanDeviceEnergyModel » peut utiliser l'un des modèles de sources d'énergie existants (p. ex. Li-Ion). Le concept de fonctionnement de ce modèle est illustré dans la figure 41. Après chaque opération effectuée par l'émetteur-récepteur (p. ex. Rx, Tx, Idle, sleep, etc.), la quantité d'énergie consommée est prélevée de la source d'énergie. Le calcul de cette quantité se base sur la durée de l'opération et le courant consommé selon chaque type d'opération. Ainsi, nous pouvons différencier, par exemple, entre le courant consommé pendant la période de sommeil et celui consommé au cours de l'émission afin d'obtenir des mesures de la consommation d'énergie plus précises. Lorsque la pile se vide (la tension de sortie de la pile est inférieure à la tension d'alimentation requise), un message de notification se déclenche et le nœud correspondant ne pourra ni transmettre ni recevoir des paquets. Ainsi, le nœud en question est considéré comme un nœud mort.

Les valeurs par défaut du courant consommé pendant chaque opération sont choisies sur la base document [24]. Cependant, l'utilisateur a la possibilité de les reconfigurer par ses propres valeurs. Les nœuds peuvent se mettre en veille pour économiser l'énergie.

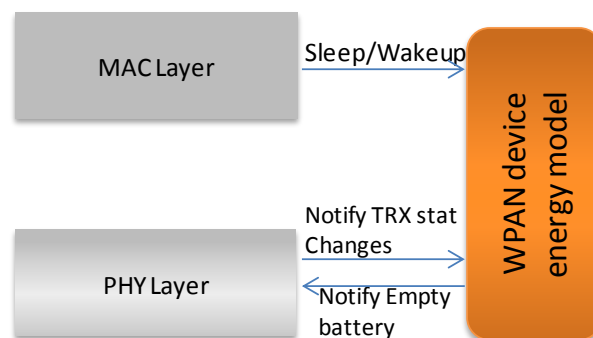


Figure 41 : Modèle d'énergie d'un nœud WPAN 'WPAN Energy Device Model'

4.2.8. Traces de simulation

L'objectif d'un simulateur étant de collecter des informations sur le fonctionnement du réseau à l'issue de la simulation, NS-3 offre plusieurs possibilités de traçage et de logging (c.-à-d. historique des événements). Dans notre modèle nous avons tiré profit de toutes ces possibilités de traçage en intégrant les deux possibilités. Premièrement, nous avons développé des assistants (Helpers) pour produire les différents fichiers trace au format ASCII. Néanmoins, afin d'éviter de ralentir la simulation et de réduire la taille des fichiers de traces dans les simulations lourdes, l'utilisateur a la possibilité de spécifier le type d'informations qu'il souhaite sauvegarder parmi les choix disponibles. Par exemple, il pourra choisir de sauvegarder les traces des échanges des paquets qui proviennent des couches supérieures (MAC DATA) tout en ignorant les restes des paquets qui circulent sur le réseau (Beacon, commandes MAC, ACK ...). Deuxièmement, nous avons fourni une autre fonctionnalité permettant d'utiliser l'API pcap (Packet Capture). Cet API permet de générer des traces des paquets émis et reçus, sous un format plus facile à lire et permettant d'analyser le détail des paquets d'une façon plus claire. Les traces générées sont compatibles avec les outils d'analyse des paquets existants. Un exemple de trace pcap visualisé dans l'outil « Wireshark » est illustré à la figure 42.

Finalement, nous avons utilisé les interfaces disponibles sous NS-3 afin de générer des traces compatibles avec l'animateur Open Source « NetAnim » [111]. L'importance de ce format, est qu'il permet l'analyse du comportement du réseau d'une manière visuelle. De plus, l'outil de visualisation est très utile pour des fins pédagogiques. La figure 43 fournit un exemple d'un réseau en étoile constitué de 64 nœuds.

6	1.006400	fe80::fdff:ff:fe00:ff02::2	ICMPv6	Router solicitation
7	1.410656	00:00:00:00:00:00:00:00	IEEE 802.15.4	Association Request
8	1.467784	00:00:00:00:00:00:00:00	IEEE 802.15.4	Data Request
9	1.470600		IEEE 802.15.4	Ack
10	2.952360	fe80::fdff:ff:fe00:ff02::1:ff00:1	ICMPv6	Neighbor solicitation
11	2.957800	fe80::fdff:ff:fe00:ff02::1:ff00:1	ICMPv6	Neighbor solicitation
12	3.005416	0x0002	IEEE 802.15.4	Data Request
13	3.013032	fe80::fdff:ff:fe00:fe80::fdff:ff:fe00:1	UDP	Source port: 61617
14	3.018152	fe80::fdff:ff:fe00:fe80::fdff:ff:fe00:1	UDP	Source port: 61617
+ Frame 13: 68 bytes on wire (544 bits), 68 bytes captured (544 bits)				
- IEEE 802.15.4 Data, Dst: 0x0001, Src: 0x0002				
- Frame Control Field: Data (0x8841)				
.... .001 = Frame Type: Data (0x0001)				
.... .0... = Security Enabled: False				
.... .0.... = Frame Pending: False				
.... .0.... = Acknowledge Request: False				
.... .1.... = Intra-PAN: True				
.... 10... = Destination Addressing Mode: Short/16-bit (0x0002)				
..00.... = Frame Version: 0				
10.... = Source Addressing Mode: Short/16-bit (0x0002)				
Sequence Number: 33				
Destination PAN: 0x0000				
Destination: 0x0001				
Source: 0x0002				
FCS: 0xfe1b (Correct)				
- 6LoWPAN				
- HC1 Encoding				
0100 0010 = Pattern: Header compression (66)				
1... = Source prefix: Compressed				
.1... = Source interface: Compressed				
..1. = Destination prefix: Compressed				
...1 = Destination interface: Compressed				
.... 1... = Traffic class and flow label: Compressed				
.... .01. = Next header: UDP (0x01)				
.... ...1 = More HC bits: True				
- HC_UDP Encoding				
1... = Source port: Compressed				
.1... = Destination port: Compressed				
..1. = Length: Compressed				
Hop limit: 64				
Source: fe80::fdff:ff:fe00:2 (fe80::fdff:ff:fe00:2)				
Destination: fe80::fdff:ff:fe00:1 (fe80::fdff:ff:fe00:1)				
Source port: 61617				
Destination port: 61630				
UDP checksum: 0xf8b7				
+ Internet Protocol Version 6, Src: fe80::fdff:ff:fe00:2 (fe80::fdff:ff:fe00:2), Dst: fe80::fdff:ff:fe00:1 (fe80::fdff:ff:fe00:1)				
+ User Datagram Protocol, Src Port: 61617 (61617), Dst Port: 61630 (61630)				
+ Data (50 bytes)				
Frame (68 bytes) Decompressed 6LoWPAN header (98 bytes)				

Figure 42 : Exemple d'utilisation de Wireshark pour visualiser le paquet UDP porté par des trames IEEE 802.15.4

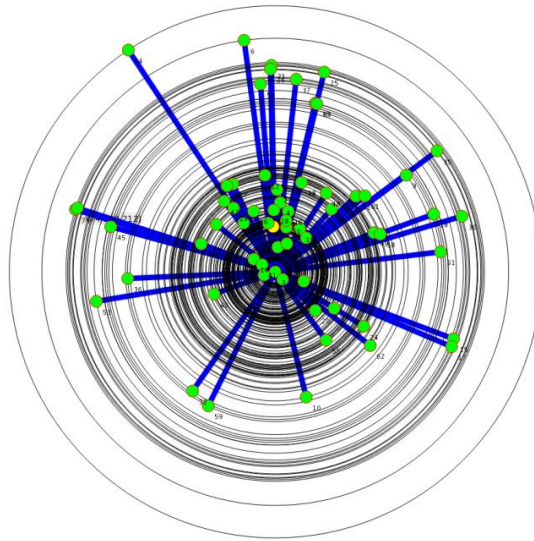


Figure 43 : Exemple de réseau WPAN visualisé à l'aide de l'animateur « NetAnim »

4.2.9. Assistants (Helpers)

Les assistants sont très utilisés par les modèles de simulation dans NS-3 afin de simplifier la configuration de la simulation et la création des nœuds. Notre modèle ne fait pas une exception, vu qu'il fournit un ensemble d'assistants permettant une configuration facile et rapide de la simulation, tout en donnant à l'utilisateur le contrôle total pour configurer toutes les composantes de ce modèle. Par exemple, des assistants pour la création des fichiers de traces, la création des nœuds, la configuration du « Network device » et la liaison des « callbacks » font partie du modèle WPAN.

En outre, nous avons développé un outil graphique permettant de générer automatiquement les scripts de simulation de manière très rapide tout en minimisant les erreurs dues à l'écriture manuelle de ces scripts. Cette interface graphique permettra aussi de créer des scénarios de simulation sans connaître tout le détail de ce standard.

4.3. Compatibilité avec le standard IEEE 802.15.4 et le modèle OSI

Le modèle proposé a été développé dans l'optique de refléter le standard IEEE 802.15.4-2006. Le modèle de base (sous NS-2) souffre du non-respect de la notion d'abstraction des couches définies dans le modèle OSI. Les différentes couches sont très dépendantes et l'accès aux données n'est pas protégé. Cela donne comme résultat une difficulté d'extension ou de réutilisation des composantes séparées. Ainsi, nous avons résolu ce problème en

utilisant le mécanisme des callbacks qui offre une indépendance totale des différents modules de ce modèle. Cette nouvelle conception permet de réutiliser ces modules en toute simplicité dans la mesure où l'on respecte les interfaces prédéfinies par le standard. Par exemple, vu que le standard IEEE 802.15.4^e utilise la même couche physique que le standard original, le module WpanPhy de notre modèle pourra être réutilisé pour une éventuelle implémentation de ce nouveau standard.

5. Évaluation de performances

L'objectif de cette évaluation de performances est de tester les fonctionnalités et les mécanismes de base proposés par notre modèle de simulation. Le tableau 9 résume les paramètres de simulation communs à tous les scénarios. D'autres paramètres, tels que la couche physique utilisée, le mode de la sous-couche MAC, la densité des nœuds et la fréquence d'envoi des données sont changés à travers les différents scénarios en vue d'évaluer les performances de ce modèle dans des conditions variées.

Pour la couche réseau, nous avons utilisé le protocole Ipv6 en plus de la couche 6LoWPAN [123]. Cette dernière permet d'adapter et compresser les données des niveaux supérieurs pour qu'elles soient compatibles aux exigences de la sous-couche MAC du réseau IEEE 802.15.4. NS-3 supporte de façon native le protocole Ipv6. Mais la couche d'adaptation 6LoWPAN n'a pas été encore disponible pendant ces travaux (version 3.16). Ainsi nous avons utilisé une implémentation 6LoWPAN développée par notre laboratoire [124] en se basant sur une implémentation se trouvant dans [125].

Paramètres	Valeurs
Topologie	Étoile
Application	UDP client/serveur
La couche réseau	Ipv6/6LoWPAN
MAC/PHY	IEEE 802.15.4
Taille des paquets (application)	50 bytes
Modèle d'affaiblissement de propagation	FriisPropagationLossModel
Modèle du temps de propagation	ConstantSpeedPropagationDelayModel

Tableau 9 : Paramètres communs de simulation

La figure 42 montre un extrait des paquets capturés et sauvegardés au format pcap. Le logiciel Wireshark est utilisé pour analyser les paquets capturés et pour décoder les champs des différents en-têtes. Cet exemple présente un paquet UDP de 50 octets transmis en utilisant le protocole Ipv6 et la couche d'adaptation 6LoWPAN. La couche MAC opère en mode beacon activé « BEn » et les nœuds associés au coordinateur utilisent l'adressage court au format 16-bits. Le calcul du CRC est activé afin de valider l'intégrité des nœuds. On peut constater que les paquets générés par notre modèle WPAN ont un format similaire aux paquets réels et sont compréhensibles par les outils standards d'analyse des paquets.

Remarque : Le modèle de Friis¹⁹ implémenté dans le simulateur NS-3 se base sur les extensions modernes du modèle original. Elles sont décrites par la formule suivante :

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2 L} \quad (X)$$

Avec :

- P_r : Puissance de réception (W)
- P_t : Puissance de transmission (W)
- G_t : Gain de transmission (sans unité)
- G_r : Gain de réception (sans unité)
- λ : Longueur d'onde (m) avec $\lambda = \frac{c}{f}$, (c est la vitesse de la lumière dans le vide (m/s) et f est la fréquence de transmission (Hz))
- d : distance (m)
- L : pertes du système (Sans unité)

5.1. Scénarios de test du modèle d'affaiblissement de propagation

Scénario 1 : échange des paquets en absence des interférences

La comparaison des modèles de propagation existants sous NS-3 est hors du cadre de cette thèse. Une étude comparative de ces modèles a été réalisée dans [126]. Néanmoins, notre modèle WPAN pourra être utilisé avec chacun de ces modèles de propagation.

¹⁹ Le modèle de propagation de Friis à été initialement décrit par « Harald T. Friis », dans "A Note on a Simple Transmission Formula".

L'objectif de ce premier scénario est de vérifier la transmission des données via le médium sans fil simulé par le « WpanChannel ». Le réseau est constitué de deux nœuds, un émetteur fixe, qui envoie de manière continue un paquet UDP de 50 octets tous les 100 ms (l'équivalent d'un débit de 4 kbps), et un nœud récepteur mobile qui s'éloigne du premier à une vitesse constante de 0.1 mètre/s (le modèle de mobilité utilisé est « ConstantSpeedMobilityModel »). La sous-couche MAC opère dans le mode sans Beacon (Beacon-less) afin d'éviter les interruptions dues aux problèmes d'association, à la transmission du Beacon ou la période de sommeil. La couche physique utilisée est « PHY 2.4 GHz » avec une sensibilité du récepteur de -85 dBm et une puissance de transmission de 1 mW (0 dBm). La figure 44 montre l'évolution de la puissance du signal reçu en fonction de la distance qui sépare la source et la destination. On constate que la puissance de signal reçu mesuré décroît avec l'éloignement du récepteur de la source. Pourtant, les paquets restent détectables jusqu'à environ 82 mètres où la valeur de la puissance du signal devient en dessous du seuil de sensibilité de réception. Cette distance représente le rayon de couverture du nœud émetteur. Ce rayon est, en effet, affecté par trois paramètres : puissance de transmission, sensibilité du récepteur et le modèle de propagation utilisé.

Pour le même scénario précédent, nous avons mesuré cette fois les valeurs de la qualité du lien « LQI (Link Quality Indicator) » au lieu de la puissance du signal reçu. Nous pouvons constater dans la figure 45 que la valeur du LQI reste constante et égale à la valeur maximale « 255 » jusqu'à une distance de 60 mètres. Au-delà de cette distance, le LQI commence à se dégrader progressivement jusqu'à une distance de 82 mètres. À cette distance, le LQI atteint la valeur 128 où le signal n'est plus reçu correctement (indétectable dans ce cas). Cette valeur pourra être considérée comme la qualité de lien minimale acceptée pour une bonne réception des trames.

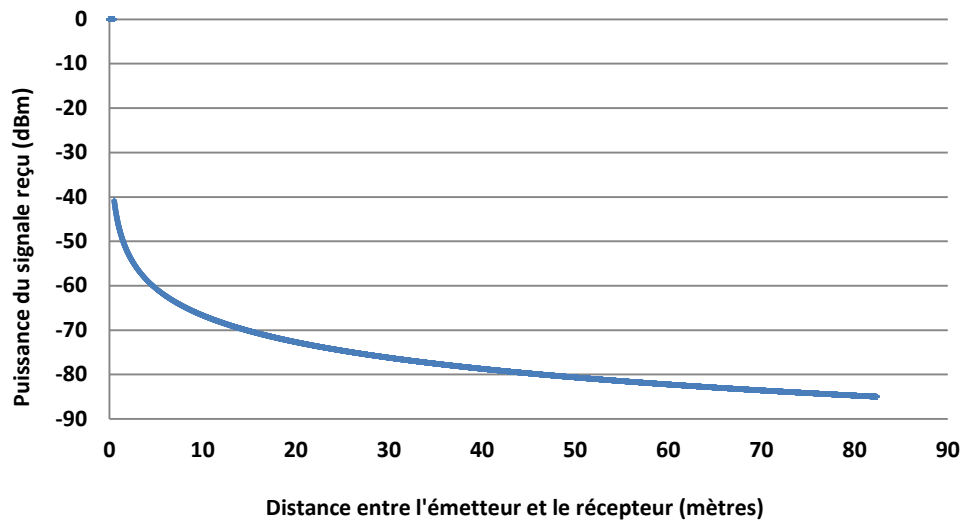


Figure 44 : Puissance reçue en fonction de la distance « source-destination » sans interférences (2 nœuds, modèle de propagation Friis)

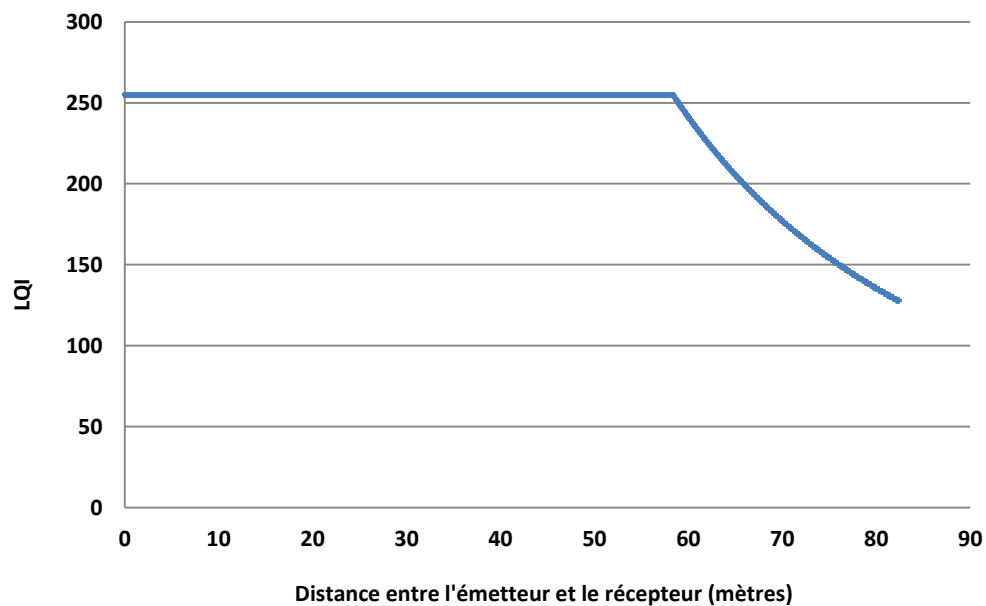


Figure 45 : LQI en fonction de la distance « source-destination » sans interférences (2 nœuds, modèle de propagation Friis)

Scénario 2 : les échanges des paquets en présence des interférences

Afin de voir l'effet des interférences sur la qualité de transmission, nous avons simulé un réseau de 19 nœuds fixes qui transmettent des paquets vers un nœud mobile. Nous avons gardé les mêmes paramètres de simulation que le scénario précédent.

La courbe de la figure 46 montre que la puissance reçue par le nœud mobile suit la loi d'atténuation due à la propagation selon le modèle de Friis. Néanmoins, si nous comparons ces résultats avec ceux du premier scénario, nous constatons que la valeur du RSSI devient plus importante pour les mêmes distances dans certaines positions. Ceci est dû à l'influence de l'interférence (bruit) sur le signal original. En effet, la puissance du signal reçu (qui n'est autre qu'une superposition de tous les signaux transmis par les émetteurs) pourra être supérieure au signal utile et la puissance du bruit peut, dans certains cas, être supérieure à celle du signal utile. Cela confirme la vérité qui postule que la puissance du signal seule n'est pas un bon indicateur de la qualité du lien.

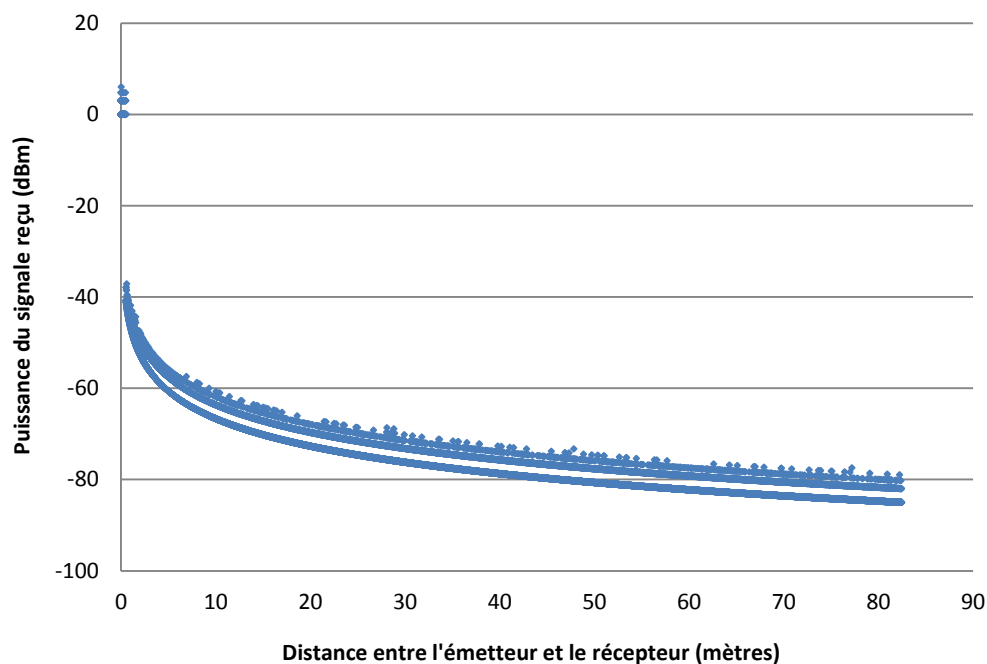


Figure 46 : Puissance reçue en fonction de la distance « source-destination » en présence des interférences (20 nœuds, modèle de propagation Friis)

Par contre, le LQI pourra être un meilleur indicateur de la qualité du lien. En effet, pour l'estimation du niveau de qualité, cet indicateur utilise deux mesures : la puissance du signal reçu et le rapport signal sur bruit (SNR « Signal to Noise Ratio »). Le couplage de ces deux mesures permet de donner une meilleure estimation de cette qualité. Ceci est bien clair dans les résultats obtenus dans la figure 47, qui montre que, contrairement à la puissance du signal, les niveaux LQI des signaux reçus sont altérés par les interférences. En effet, si on le compare avec celui du premier scénario, on remarque une fluctuation des valeurs du LQI, même pour de faibles

distances. Ce comportement vient du rapport SNR qui devient plus faible en présence des interférences.

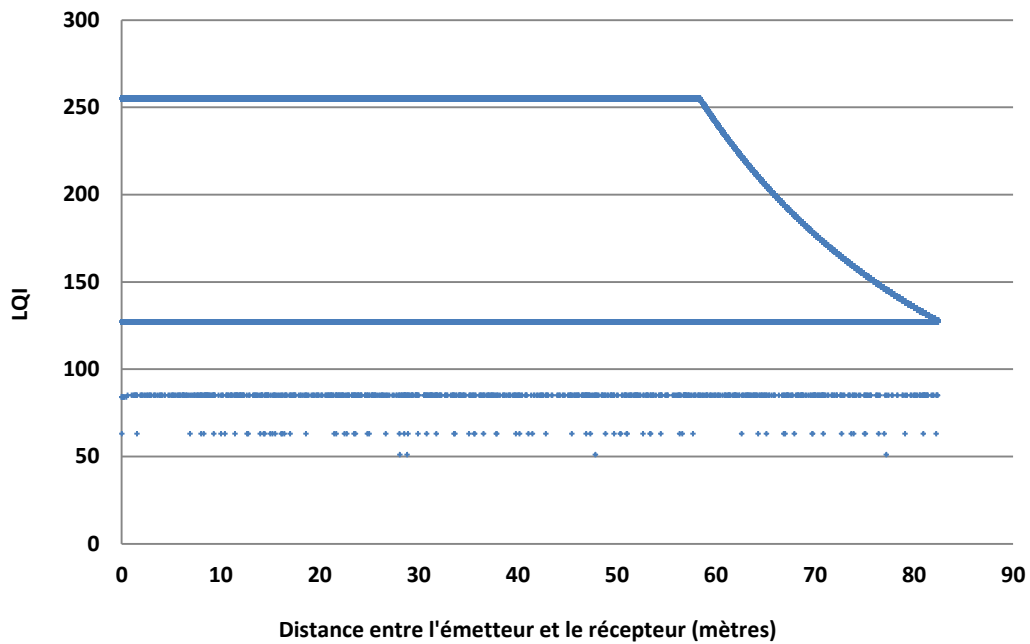


Figure 47 : LQI en fonction de la distance « source-destination » en présence des interférences (20 nœuds, modèle de propagation Friis)

On peut déduire de ces simulations que l'indicateur de la qualité du lien « LQI » est plus précis pour estimer la qualité des liaisons entre les nœuds que la puissance du signal seule.

5.2. Mesure du délai pour les différentes couches MAC

Scenario 1 : mode MAC Sans Beacon « Beaconless »

L'objectif de cette partie est de mesurer, dans le mode MAC sans beacon, l'influence de la densité des nœuds (allant de 2 à 64 nœuds) sur le délai pour les différentes couches physiques du modèle WPAN. Les nœuds du réseau sont fixes et tous les nœuds sources envoient des paquets UDP, tous les dixièmes de secondes, vers un puits « Sink » (un débit de 4 kb/s pour chaque nœud).

Les résultats obtenus dans la figure 48 montrent que le délai de communication est affecté par trois facteurs. Le premier est la densité des nœuds qui a une influence considérable sur le temps d'accès au médium. En effet, le médium devient plus occupé avec l'augmentation du nombre de nœuds qui doivent attendre plus longtemps avant d'accéder au canal (à cause de l'augmentation de l'exposant du backoff). En outre, en cas d'envoi des paquets, la probabilité

d'avoir des collisions augmente et, par conséquent, le nombre des retransmissions augmente. Le deuxième facteur est la bande passante offerte par la couche physique utilisée. Lorsque la bande passante n'est pas suffisante, les contentions deviennent encore plus rudes ce qui augmentera naturellement le délai de communication. Ceci est bien visible dans les courbes de la figure 48, où le délai mesuré pour les couches physiques qui offrent, selon le standard, un débit de 250 kb/s (2.4 GHz O-QPSK, 915 MHz ASK, 915 MHz O-QPSK et 868MHz ASK), est inférieur à celui des autres couches physiques qui offrent seulement 100, 40 ou 20 kb/s pour des nombres de nœuds identiques. Le dernier facteur est le débit de symbole qui a une influence plutôt sur les mécanismes internes de la sous-couche MAC. En effet, les différents calculs au sein de la sous-couche MAC sont donnés en nombre de symboles, comme il est le cas du backoff utilisé par l'algorithme CSMA-CA. Pour cet exemple, le temps d'attente aléatoire devient plus grand pour les faibles débits de symboles. Cette situation est bien soulignée dans le cas de la couche PHY 868MHz ASK où, même avec un débit binaire de 250 kbps (similaire aux couches PHY 2.4 GHz O-QPSK, 915 MHz ASK, 915 MHz O-QPSK), le délai pour le premier est supérieur à celui des derniers. Ce comportement vient du faible débit de symboles offert par la couche PHY 868 MHz ASK (12.5 ksymbole/s) comparé aux autres versions (50 à 62.5 ksymbole/s).

Le délai de propagation²⁰ est relativement négligeable, car les rayons de couverture sont très limités (quelques dizaines de mètres).

²⁰ On rappelle que le modèle de propagation utilisé dans tous les scénarios est : ConstantSpeedPropagationDelayModel

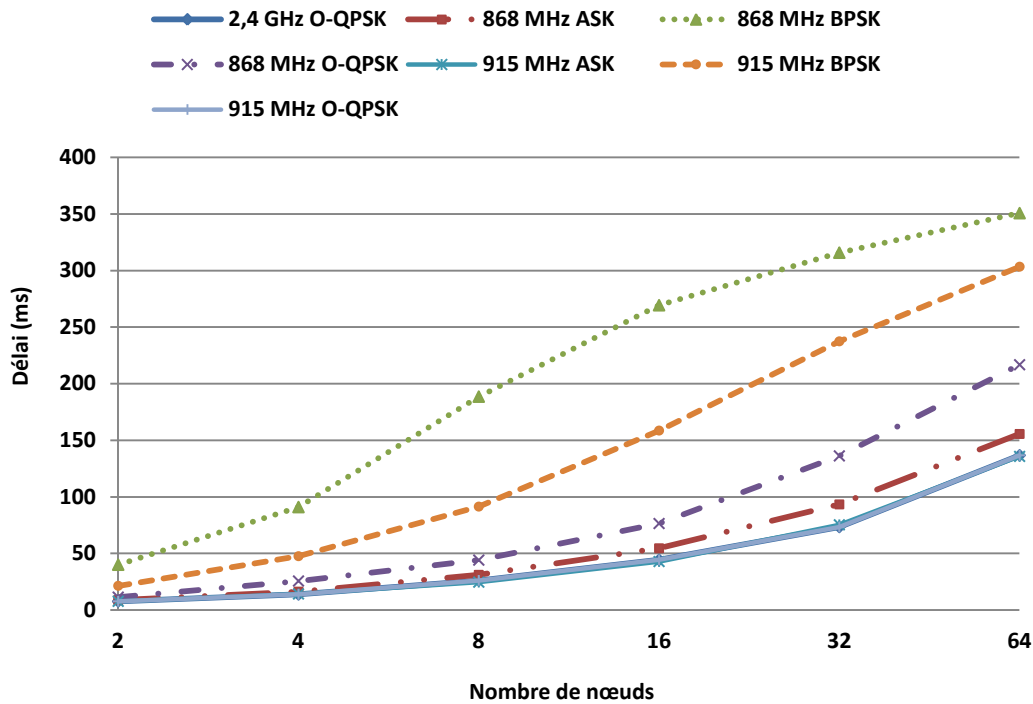


Figure 48 : Délai en fonction du nombre pour les différentes couches PHY

Scénario 2 : le mode MAC avec beacon activé « Beacon-enabled »

Dans ces scénarios, nous évaluons l'influence des paramètres BO et SO, et plus particulièrement le rapport cyclique, sur le délai de communication. Ce délai est mesuré pour les différents nombres de nœuds. Les paramètres de transmission sont pareils aux scénarios précédents (l'envoi des paquets de plusieurs sources vers un puits à un débit de 4 kb/s). Le paramètre BO est fixé à la valeur « 8 » alors que SO varie de « 2 » à « 8 ». La figure 49 montre l'impact de la valeur du rapport cyclique sur le délai. Ce dernier augmente lorsque le temps de sommeil devient plus important, car les données non transmises dans la période active restent stockées dans les files d'attente des nœuds pendant tout le temps de sommeil. Ainsi, on peut réduire la consommation de l'énergie en augmentant le temps de sommeil. Cependant, cela se fera au détriment du délai de communication. Le compromis énergie/délai est l'un des plus grands problèmes à résoudre dans le cadre des RCSF, notamment, dans les applications où la réduction du délai de communication est aussi importante que l'économie d'énergie. Ainsi, lors de la conception d'un réseau de capteurs sans fil, il faut choisir soigneusement le débit (la couche

PHY) et le rapport cyclique (BO et SO) en fonction des contraintes des capteurs et des besoins des applications en termes de qualité de service.

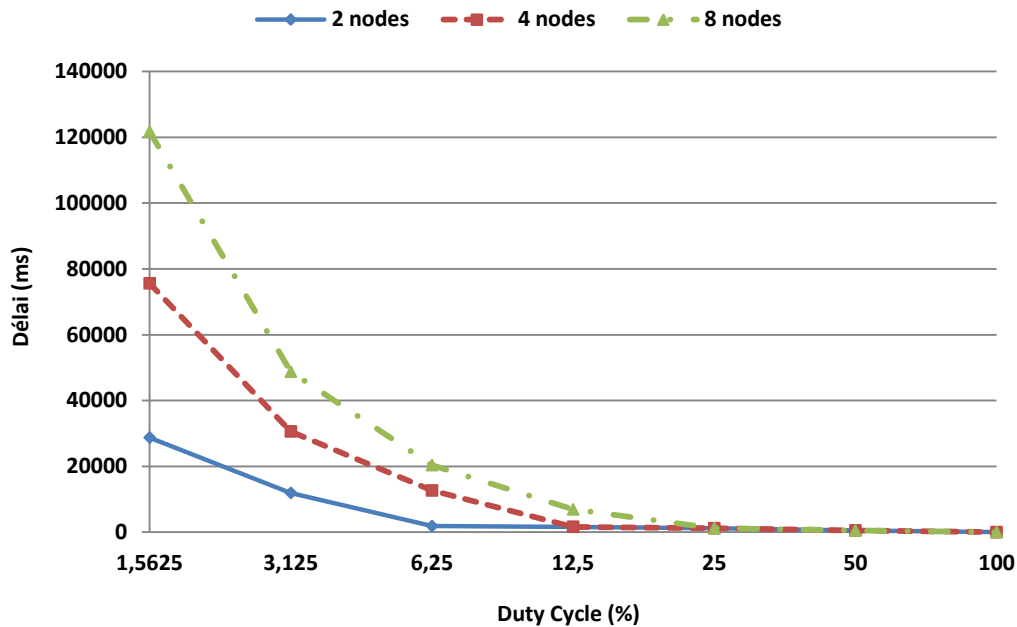


Figure 49 : Délai en fonction du « Duty Cycle » pour différentes densités

5.3. Mesure du débit pour les différentes couches physiques

Le standard IEEE 802.15.4 étant conçu pour les réseaux à faible taux de données « LR-WPAN » (les RCSF par exemple), il propose un ensemble de couches physiques avec des bandes passantes allant jusqu'à 250 kbps²¹. L'objectif de cette partie est d'évaluer le débit pour toutes ces couches physiques et l'utilisation effective de la bande passante dans des conditions variées.

Nous commençons par la mesure du débit utile en utilisant la couche PHY 2.4 GHz. Cette couche PHY offre un débit maximal théorique de 250 kbps. On augmente progressivement le taux de données au niveau de la couche application et on mesure le débit utile. On constate dans la figure 50 une croissance de ce dernier avec l'augmentation du taux d'envoi des paquets UDP jusqu'à un débit de 56 kbps. Au-delà de cette limite, le débit utile ne suit plus le taux

²¹ Dans certaines implémentations réelles, cette bande passante peut atteindre 2 Mb/s (paragraphe 2.3.1 du chapitre III)

d'envoi des données et reste figé dans des valeurs proches de 56 kbps. Cette perte de bande passante est en effet due à plusieurs facteurs, entre autres ; les en-têtes ajoutés aux données utiles (entête MAC, Ipv6 ...), l'algorithme CSMA-CA (backoff et CCA) et les espacements inter-trame « IFS (Inter-Frame Spacing) ».

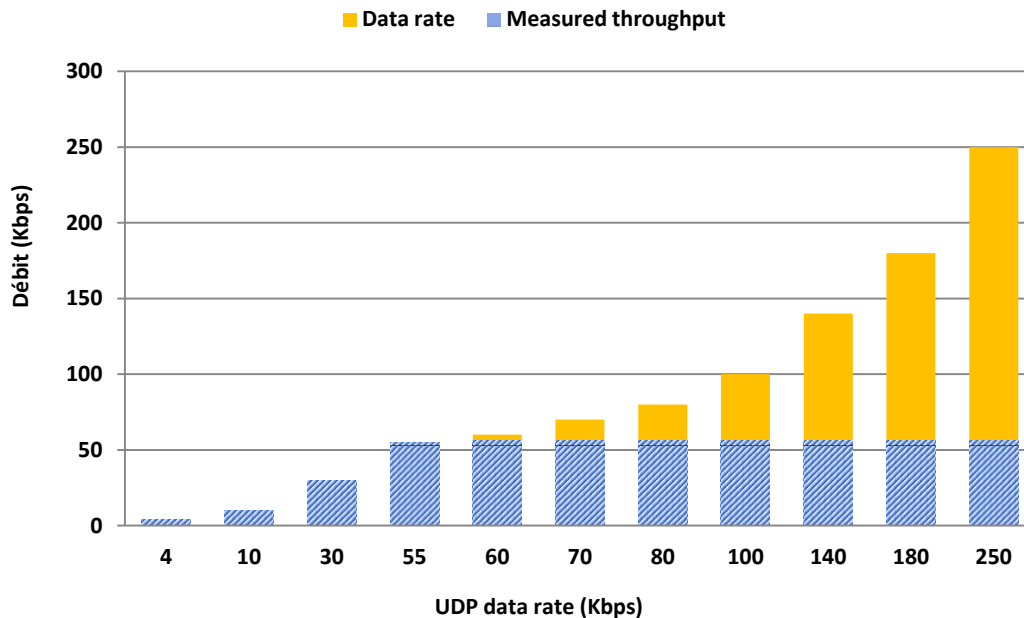


Figure 50 : Débit utile en fonction des taux d'envoi des paquets UDP

Dans la figure 51, on compare le débit maximal effectif, mais cette fois pour les différentes couches physiques du standard. Nous avons mesuré le débit utile (50 octets/paquet de données UDP) représenté par les barres bleues et le débit des trames MAC (charge utile + les différents en-têtes) représenté par les barres vertes. Les barres rouges correspondent à la bande passante théorique de chaque variante de la couche physique. L'objectif est de voir l'influence des en-têtes sur l'efficacité d'utilisation de la bande passante. Le comportement est identique à celui de la figure 50 qui montre qu'un nœud seul ne peut pas occuper toute la bande passante. En outre, en comparant les barres bleue et verte, nous constatons qu'une partie de la bande passante est consommée par la surcharge qui contient les informations des différentes couches. Au niveau de la couche MAC, nous avons utilisé le mode Beaconless ce qui introduit des adresses MAC longues de 64-bits. Cette surcharge pourra être réduite lors de l'utilisation des modes avec ou sans beacon activé, notamment pour les nœuds qui se sont associés correctement au coordinateur. Ces nœuds peuvent utiliser des adresses MAC courtes de 16-bits. Même remarque pourra être appliquée pour les en-têtes Ipv6 et UDP. Dans nos scénarios, nous avons utilisé des techniques

de compression pour réduire la taille de la surcharge « Overhead » grâce à la couche d'adaptation 6LoWPAN. En effet, l'utilisation du protocole Ipv6 introduit des adresses sources et destinations très longues (128-bits chacune). La taille de ces en-têtes peut être, dans certains cas, plus grande que la charge utile. La couche 6LoWPAN permet de réduire l'en-tête Ipv6 et UDP à une taille très petite.

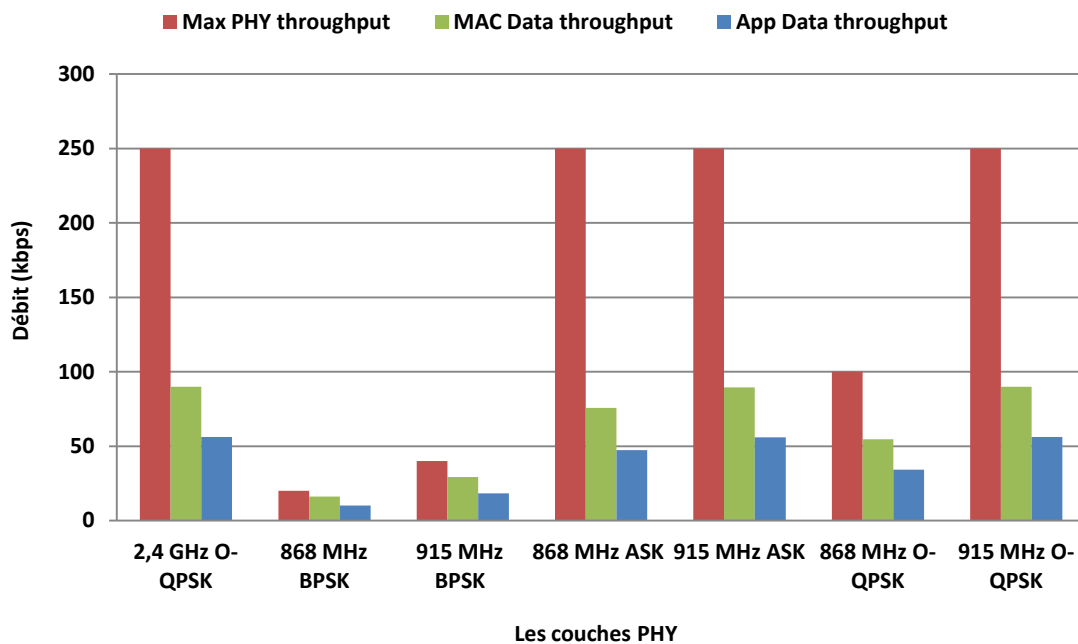


Figure 51 : Débits utiles mesurés pour les différentes couches physiques

Dans la figure 52, nous avons mesuré le débit en fonction du nombre de nœuds. Bien évidemment, l'utilisation de la bande passante pour chaque nœud diminue avec l'augmentation du nombre de concurrents dans le canal. Le passage à l'échelle introduit plus d'interférences et, par conséquent, une difficulté d'accès au canal (échec du CCA), des retransmissions des paquets, des backoffs qui deviennent plus longs, etc.

Ainsi le débit mesuré est également influencé par les paramètres de la couche physique, plus particulièrement, le débit binaire et le débit de symboles. En effet, le débit de symboles a un impact sur le backoff utilisé par l'algorithme CSMA-CA. La figure 52 fournit un exemple de ce problème dans lequel les couches PHY 868 MHz ASK et 868 MHz O-QPSK offrent le même débit binaire, alors que le débit mesuré pour les deux couches n'est pas le même. Par contre, les courbes des débits mesurés pour les couches PHY 915 MHz O-QPSK et 2.4 GHz O-QPSK sont confondues, car ils offrent les mêmes débits binaires et débits de symboles.

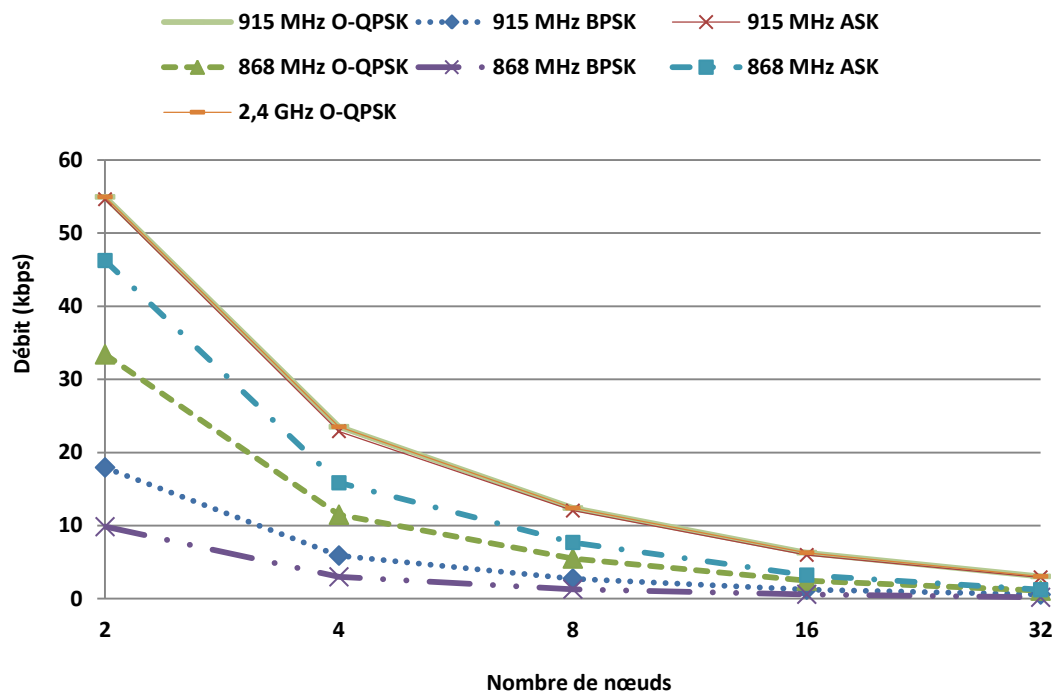


Figure 52 : Débits binaires mesurés en utilisant les différentes couches physiques du standard IEEE 802.15.4 pour des nombres de nœuds différents

Remarques générales :

Les résultats obtenus montrent que le délai et le débit sont influencés par le changement du nombre de nœuds (la densité) surtout dans les réseaux denses. Le standard offre une solution pour résoudre ce problème en utilisant le mécanisme GTS. Ce mécanisme n'est pas encore complété dans le modèle WPAN. En effet, jusqu'à présent, nous n'avons implémenté que la partie de gestion des commandes GTS, le reste sera complété ultérieurement.

Nous avons utilisé les adresses MAC au format 48-bits au lieu de 64-bits, car la version actuelle du modèle du protocole Ipv6 sous NS-3 ne supporte que le premier format.

Les fonctions de sécurité ne sont pas encore supportées et la version 6LoWPAN utilisée ne supporte pas la compression des adresses Ipv6 globales (IPHC and NHC).

6. Conclusion du chapitre

Dans ce chapitre nous avons présenté un nouveau modèle de simulation du standard IEEE 802.15.4-2006 pour le nouveau simulateur NS-3. Ce modèle offre une implémentation de la majorité des mécanismes et des fonctionnalités proposées dans ce standard, entre autres, les différents modes de la sous-couche MAC et toutes les couches physiques spécifiées dans ce standard. L'un des objectifs de ce modèle était de respecter le concept d'abstraction des couches du modèle OSI. Dans le modèle WPAN proposé, les communications entre les différentes couches se font de manière exclusive à travers des interfaces standardisées appelées « des points d'accès au service (SAP) » à l'aide du mécanisme des callbacks.

Pour l'extraction des résultats de simulation, nous avons mis à disposition tous les formats de traçage offerts par le simulateur NS-3, à savoir, le format de trace ASCII, le format PCAP et le format « xml » pour l'animateur « NetAnim ».

La configuration de simulation est aussi simplifiée grâce aux assistants « Helpers ». En outre, nous avons développé une application graphique permettant de générer automatiquement les scripts de simulation de manière très rapide tout en minimisant les erreurs dues à l'écriture manuelle de ces scripts.

L'évaluation de performances réalisée montre que ce modèle donne les résultats prévus à travers les différents scénarios étudiés. Pour l'instant nous avons validé le modèle que dans un réseau à un saut (étoile).

Nous sommes convaincus que ce travail peut être vu comme un fondement qui facilitera les travaux futurs sur les technologies I/WoT sous le simulateur NS-3.

Conclusion Générale et Perspectives

1. Conclusion Générale

L'assurance de la qualité de service dans le domaine des réseaux de capteurs sans fil représente un défi de recherche très important, plus particulièrement dans ce type de réseaux caractérisé par les contraintes intrinsèques des nœuds RCSF, et dans lesquels l'économie d'énergie est une question axiale. Pour économiser l'énergie et prolonger la durée de vie du réseau, les protocoles de communication existants pour les RCSF se basent sur le concept du rapport cyclique « Duty Cycle », dans lequel, les nœuds se mettent en veille de manière périodique. Cependant, ce mode d'opération réduit aussi la réactivité des nœuds et, par conséquent, altère le niveau de qualité de service. Ceci constitue un compromis difficile à assurer dans les scénarios où la garantie de la qualité de service est aussi importante que l'économie d'énergie.

Dans le cadre de cette thèse, nous nous sommes focalisés sur les mécanismes de la qualité de service proposé par le standard IEEE 802.15.4. Nous avons étudié plusieurs protocoles MAC pour les réseaux de capteurs sans fil, plus particulièrement ceux basés sur ce standard. Pour améliorer la qualité de service, ces approches proposent des mécanismes qui peuvent être répartis en trois catégories : avec contention, sans contention et hybrides. Nous avons identifié des limitations auxquelles il n'y avait pas encore de solutions.

L'approche que nous avons proposée présente une solution aux problèmes identifiés, notamment pour dans le cadre du modèle de communication multi-sources multi-sink. Le nouveau protocole « FF-MAC » permet de résoudre l'ensemble de ces problèmes et de permettre aux utilisateurs de réduire l'activité des nœuds en réduisant le rapport cyclique, tout en assurant des performances élevées pour les trafics temps-réel.

Notre réflexion s'est accompagnée d'une démarche de simulation en vue d'évaluer les performances de notre approche. Nous avons dû compléter le modèle existant par le mécanisme des GTS défini dans le standard IEEE 802.15.4. Ensuite, nous avons implémenté les modules essentiels au fonctionnement du protocole FF-MAC, notamment le gestionnaire de la trame

PRTPA, et la gestion de la période D-CFP. Les résultats présentés confirment les améliorations d'un point de vue théorique.

Afin de valider le fonctionnement de notre protocole dans un environnement réel, nous l'avons implémenté sur des nœuds RCSF réels. Les résultats expérimentaux sont satisfaisants et convergent avec les résultats de simulation. Ce qui permet à la fois de valider le modèle de simulation proposé et de montrer que le protocole est utilisable dans des applications réelles.

De plus, le protocole proposé est compatible avec la plupart des approches existantes, notamment celles que nous avons présentées dans l'état de l'art, car elles se basent sur des principes complémentaires.

À côté de ces travaux, nous avons proposé une implémentation du standard IEEE 802.15.4-2006 sous le simulateur NS-3. Ce travail est très important pour les chercheurs spécialisés dans le domaine des RCSF, qui souhaitent migrer vers ce nouveau simulateur. Nous avons adopté une implémentation proche du modèle existant dans le simulateur NS-2 afin de faciliter la migration vers ce nouveau modèle. Néanmoins, nous avons rendu son architecture plus proche du standard en permettant de respecter le modèle OSI. Cela permettra également de rendre le modèle plus facile à comprendre et à utiliser, et ses composantes peuvent être réutilisées par d'autres modèles facilement.

Les résultats obtenus sont encourageants et ouvrent plusieurs perspectives de recherche.

2. Perspectives

Notre protocole FF-MAC pourra être amélioré et mis à jour pour qu'il soit compatible avec la nouvelle version du standard (c.-à-d. IEEE 802.15.4^e) et bénéficier des nouvelles fonctionnalités. En effet, en plus du format original du standard, le nouveau standard offre un format appelé « *DSME multi-superframe* » très intéressant pour les trafics hétérogènes. Le protocole FF-MAC peut aussi être amélioré pour l'environnement multi-sauts, notamment dans le cadre de la topologie en arbre de clusters « cluster tree ». La possibilité d'indiquer les paquets temps-réel dans le paquet PRTPA pourra être utile pour les nœuds parents auxquels le coordinateur peut leur réserver des D-GTS. Ainsi, la garantie d'acheminement pourra être assurée en amont vers les autres étoiles.

Nous visons aussi l'évaluation de notre modèle dans un environnement multi-sauts par l'utilisation du protocole RPL [127][128] et le reste des protocoles de la pile de protocoles des technologies I/WoT (à titre d'exemples, le mécanisme de découverte des voisins ND pour 6LoWPAN [129] et le protocole CoAP [130]). En outre, nous allons compléter l'implémentation du mécanisme de GTS qui manque dans le modèle de base vu l'intérêt de la QoS dans plusieurs applications et les demandes que nous avons reçues d'autres chercheurs. Une autre perspective importante se manifeste dans la validation du modèle proposé par rapport à des réseaux de capteurs réels en utilisant les plateformes de capteurs sans fil (iLive par exemple [131][132]).

Grâce au mécanisme des callback, la réutilisation des différentes couches du modèle WPAN est très facile. Dans ce cadre une implémentation du standard IEEE 802.15.4^e pourra être effectuée en offrant une implémentation des nouveaux mécanismes de la couche MAC tout en réutilisant la couche physique existante. En effet, le nouveau standard a conservé l'utilisation de la couche PHY existante du standard IEEE 802.15.4-2006 pour des raisons de compatibilité avec les capteurs sans fil existants.

Appendice

Annexe A :

Quelques observations expérimentales

L'utilisation de l'algorithme CSMA/CA peut donner des résultats identiques à celles obtenues par l'utilisation des GTS. Ceci se manifeste dans un cas spécial quand la densité des nœuds et la quantité du trafic du réseau sont faibles. Dans ces conditions quasiment tous les nœuds vont réussir à envoyer leurs données, et par conséquent, il n'y aura pas une grande différence entre l'envoi dans la CAP et l'envoi dans la CFP. Cependant, dans ces mêmes conditions, les performances moyennes de la méthode CSMA-CA peuvent être meilleures que ceux lors de l'utilisation des GTSs. En effet, selon le standard IEEE 802.15.4, dans chaque Beacon envoyé, le coordinateur annonce jusqu'à 7 adresses MAC ayant des données en attente. Si un des nœuds émetteurs envoie de nouveaux paquets, ces derniers seront misent dans la file d'attente des données indirectes « Indirect Queue » du coordinateur. Si l'adresse du nœud destination est indiquée dans la liste annoncée par le coordinateur dans son Beacon, et s'il envoie une commande MAC de demande de données « DATA REQUEST », après l'envoi de nouvelles données par la source. Ce nœud aura la possibilité de récupérer les anciens paquets transmis dans les supertrames précédentes en plus des nouveaux paquets (envoyées pendant la même CAP). Deux cas de figure peuvent avoir lieu :

Si l'on permet l'envoi des données temps réel dans la CAP par les nœuds ayant réussi à réserver un GTS. Ces nœuds essayeront de transmettre leurs données dans la CAP, et s'ils échouent. Ils les transmettront dans leurs GTSs respectifs. L'avantage de cette approche est que ces nœuds auront la possibilité d'envoyer les données rapidement dans la CAP sans attendre son GTS. De plus ces nœuds seront sur de pouvoir envoyer les données temps réel au coordinateur dans la même supertrame. Les performances seront légèrement meilleures que la CSMA-CA, qui peut ne pas permettre l'envoi de tous les paquets dans la même CAP. Cette méthode présente un grand inconvénient lié au gaspillage de la bande passante vu que la plupart des paquets seront transmis dans la période CAP (vu la faible densité, et par conséquence, la faible probabilité de collisions). Le GTS sera peu utilisé. Ainsi, la bande passante réservée sera peu utilisée, et les nœuds temps réel consommeront la bande passante des autres nœuds dans la période CAP.

La seconde approche consiste à envoyer les données des nœuds ayant des GTS uniquement dans leur GTS respective. Cela évitera le gêne introduit par la 1^{ère} approche dans la CAP, et d'utiliser les slots réservés de manière plus efficace. Cependant, les données temps réel doivent attendre l'arrivée du GTS pour effectuer des transmissions ce qui introduit un délai supplémentaire. La possibilité de pouvoir envoyer/recevoir les nouveaux paquets transmis dans la même supertrame n'est plus possible car les nouveaux paquets sont envoyés après la réception des anciens paquets par leurs destinataires respectifs. Par contre, cette possibilité reste valable pour les données envoyées en CAP. Donc le délai moyen de bout-en-bout sera plus faible pour les transmissions en CAP que celui en CFP. Mais, comme nous l'avons déjà mentionné, cela reste valable seulement dans des réseaux de faible densité des nœuds et de faible trafic réseau.

Annexe B :

Implémentation du mécanisme PRTPA

Le mécanisme PRTPA est au cœur de notre protocole FF-MAC. Ce mécanisme est responsable de la création de la période D-CFP, de l'ordonnancement des slots D-GTS et de la création et transmission de la trame PRTPA. Les informations sur la période D-CFP, à savoir : les adresses des nœuds propriétaires des D-GTS, l'ordonnancement de ces derniers et la taille de la D-CFP qui détermine le début de la CAP, sont diffusé dans la trame PRTPA. L'implémentation de ce mécanisme au sein de la pile « Atmel MAC Software for IEEE 802.15.4 transceivers » se trouve dans un nouveau fichier dédié appelé « mac_prtpa.c ». Ce fichier se trouve dans l'emplacement : « MAC_v_2_8_0\MAC\Src\mac_prtpa.c »

1. Dans le coordinateur.

```
/**
 * @brief Appends the pending RT short addresses into the PRTPA frame
 *
 * This function appends the pending RT short addresses to the PRTPA frame
 * based on frames currently in the rt indirect queue.
 * Only frames sent in GTS transmit are appended
 *
 * @param buf_ptr Pointer to the indirect rt data in the rt indirect queue
 * @param handle Callback parameter
 * @return 0 to traverse through the full rt indirect queue
 */
static uint8_t add_pending_rt_short_address_cb(void *buf_ptr, void *index)
{
    frame_info_t *frame = (frame_info_t *)buf_ptr ;

    /* Only 7 short addresses are allowed in one PRTPA frame. */
    if (pending_rt_address_count < PRTPA_MAX_PEND_ADDR_CNT)
    {
        /*
         * Only if the destination addressing mode is short address mode
         then the
         * indirect data is used to populate the PRTPA buffer with short
         * destination address.
         */
        if (FCF_SHORT_ADDR ==
            ((frame->mpdu[PL_POS_FCF_2] >> FCF_2_SOURCE_ADDR_OFFSET) & FCF_ADDR_MASK))
        {
            /* Get Real source short Address From MSDU */

```

```

uint16_t short_addr =
    convert_byte_array_to_16_bit(
        &frame->mpdu[PL_POS_MSDU_SRC_ADDR_START]);
if (short_addr ==
    mac_gts_db.spec.fields.list[(uint8_t *)index].devAddr16)
{
    /* Get destination short Address From MSDU */
    short_addr =
        convert_byte_array_to_16_bit(
            &frame->mpdu[PL_POS_DST_ADDR_START]);
    /* save Short Address in D-GTS database */
    d_gts_db[pending_rt_address_count].destAddr = short_addr;
    /* Copy Short Address */
    memcpy(prtpa_ptr, &short_addr, sizeof(uint16_t));
    prtpa_ptr += sizeof(uint16_t);

    /* D-GTS Starting Slot */
    *prtpa_ptr = pending_rt_address_count & 0x0F;
    /* D_GTS Length */
#ifdef ACCURATE_D_GTS
    uint8_t frame_size = 28; //Bytes (23B « MAC » + 5B
« PHY »)

    132um132ript_on = true;
    uint8_t numNodes = 1;
    d_gts_db[pending_rt_address_count].length =
        numNodes * frame_size * 2;
    d_gts_db[pending_rt_address_count].length +=

(numNodes-1) * (frame_size<= aMaxSIFSFrameSize) ? macMinSIFSPeriod_def :
macMinLIFSPeriod_def;
    if (ack_on)
    {
        uint8_t ack_symb = (aTurnaroundTime + aUnitBackoffPeriod + 20);
        d_gts_db[pending_rt_address_count].length += ack_symb*numNodes;
    }
    d_gts_db[pending_rt_address_count].length /= aUnitBackoffPeriod;
    d_gts_db[pending_rt_address_count].length += 1;
#else
    d_gts_db[pending_rt_address_count].length = 1;
#endif /* ACCURATE_D_GTS */

    *prtpa_ptr++ |= d_gts_db[pending_rt_address_count].length << 4;
    pending_rt_address_count++;

    return 1;
}
}
return 0 ;
}

```

1.1. Création du descripteur des D-GTS

```

void build_d_gts_descriptors(void)
{
    search_t find_buf;

```

```

/* save the ptr to gts specification byte */
/* Initialize extended address count. */
pending_rt_address_count = 0;
/*
 * This callback function traverses through the rt indirect queue and
 * updates the beacon buffer with the Real-Time pending short
addresses.
 */
find_buf.criteria_func = add_pending_rt_short_address_cb;
/*
 * At the end of this function call (qmm_queue_read), the PRTPA buffer
 * will be updated with the short address (if any) of the rt indirect
 * data (if any) present in the rt indirect queue and that was sent in
 * previous GTS Transmit.
 */
for (uint8_t index = 0; index < mac_gts_db.spec.count; index++)
{
    if(!mac_gts_db.spec.recvOnly[index])
    {
        find_buf.handle = &index ;
        qmm_queue_read(&rt_indirect_data_q, &find_buf);
    }
}
/* Set D-GTS count */
d_gts_count = pending_rt_address_count;
}

```

1.2. Préparation et transmission du paquet PRTPA

```

/**
 * @brief Builds and transmits the beacon frame
 *
 * This function is called to build a beacon frame. For beaconless network
 * this function also transmits the generated beacon frame.
 *
 * @param beacon_enabled Flag indicating the mode of beacon transmission
 * @param beacon_buffer_header For build without beacon support only:
 *                               Pointer to buffer of beacon frame to be
 *                               transmitted in nonbeacon network.
 */
void mac_build_and_tx_prtpa_cb(void)
{
    frame_info_t *prtpa_frame ;
    uint16_t fcf ;
    uint8_t frame_len ;
    uint8_t *frame_ptr ;
    uint8_t *temp_frame_ptr ;
    /*
     * The frame is given to the TAL in the 'frame_info_t' format,
     * hence an instance of the frame_info_t is created.
     */
    prtpa_frame = (frame_info_t *)prtpa_buffer ;
    //prtpa_frame = (frame_info_t *) (BMM_BUFFER_POINTER(buffer_header));

    /* Buffer header not required in PRTPA build. */
    prtpa_frame->msg_type = PRTPA_REQUEST;

    /* D-GTS Descriptors and count */

```

```

uint8_t gts_descr_size = 0 ;
if(GTSDataBase_isThereTransmitGTS(&mac_gts_db)) // if GTS transmit
reserved
{
    uint8_t d_gts_descriptors_ptr[21] ;
    /* save the ptr to gts specification byte */
    prtpa_ptr = d_gts_descriptors_ptr ;
    build_d_gts_descriptors();

    /* Calculate MAC Command payload size */
    gts_descr_size = pending_rt_address_count * (sizeof(uint16_t)+1);

    /* Get the payload pointer. */
    frame_ptr = temp_frame_ptr =
        (uint8_t *)prtpa_frame +
        LARGE_BUFFER_SIZE -
        PRTPA_PAYLOAD_LEN - gts_descr_size
        - 2; /* Add 2 octets for FCS. */

    /* Update the payload field. */
    /* Commande ID = 0x0A */
    *frame_ptr++ = PRTPAREQUEST;

    /* D-GTS Counter*/
    *frame_ptr++ = pending_rt_address_count;//gts_descr_size;
    d_gts_count = pending_rt_address_count;

    memcpy(frame_ptr, d_gts_descriptors_ptr, gts_descr_size) ;
}
else // no need to check the rt indirect Queue when there is no GTS
transmit
{
    frame_ptr = temp_frame_ptr =
        (uint8_t *)prtpa_frame +
        LARGE_BUFFER_SIZE -
        PRTPA_PAYLOAD_LEN - 2; /* Add 2 octets for FCS. */

    /* Update the payload field. */
    /* Commande ID = 0x0A */
    *frame_ptr++ = PRTPAREQUEST;

    /* D-GTS Counter*/
    *frame_ptr++ = 0;
    d_gts_count = 0;
}

/* Update the payload length. */
frame_len = PRTPA_PAYLOAD_LEN +
    gts_descr_size + // MAC (CMD) PRTPA D-GTS Descr length
    2 + // Add 2 octets for FCS
    2 + // 2 octets for short Source Address
    2 + // 2 octets for short dest Address
    2 + // 2 octets for short Dest PAN-Id (Intra PAN)
    3; // 3 octets DSN and FCF

/* Get the payload pointer again to add the MHR. */
frame_ptr = temp_frame_ptr ;
/*
 * Source address.

```



```

    */
    frame_ptr -= 2;
    convert_16_bit_to_byte_array(tal_pib.ShortAddress, frame_ptr);

    /*
     * Dest address.
     */
    frame_ptr -= 2;
    convert_16_bit_to_byte_array(BROADCAST, frame_ptr);

    /*
     * Dest PAN-Id
     */
    frame_ptr -= 2;
    convert_16_bit_to_byte_array(tal_pib.PANId, frame_ptr);

    /* Set BSN. */
    frame_ptr--;
    *frame_ptr = mac_pib.mac_BSN - 1; //Same as Beacon : may be useful to
detect Bcn loss

    fcf = FCF_SET_FRAMETYPE(FCF_FRAMETYPE_MAC_CMD) |
          FCF_SET_DEST_ADDR_MODE(FCF_SHORT_ADDR) |
          FCF_SET_SOURCE_ADDR_MODE(FCF_SHORT_ADDR) |
          FCF_SET_D_GTS_COUNT(d_gts_count) |
          FCF_PAN_ID_COMPRESSION;

    /* Set the FCF. */
    frame_ptr -= 2;
    convert_spec_16_bit_to_byte_array(fcf, frame_ptr);

    /* First element shall be length of PHY frame. */
    frame_ptr--;
    *frame_ptr = frame_len;

    /* Finished building of frame. */
    prtpa_frame->mpdu = frame_ptr ;
    //pal_get_current_time(&prtpa_tx_time);

    retval_t status = tal_tx_frame(prtpa_frame, NO_CSMA_NO_IFS, true);
    if (MAC_SUCCESS == status)
    {
        MAKE_MAC_BUSY();
        if(d_gts_count)
        {
            d_gts_timerStep = GTS_IDLE;
            d_gts_counter = 0;
            pal_timer_start(T_CFP_USAGE_Time,
TAL_CONVERT_SYMBOLS_TO_US(2*macMinLIFSPeriod_def), //MIN_TIMEOUT
                           TIMEOUT_RELATIVE,
                           (FUNC_PTR)mac_t_d_cfp_usage_cb,
                           NULL)

        }
    }
}

```

```
/*
 * @brief Checks for matching short address
 *
 * This callback function checks whether the passed short address
 * matches with the frame in the queue.
 *
 * @param buf Pointer to rt indirect data buffer
 * @param short_addr Short address to be searched
 *
 * @return 1 if short address passed matches with the destination
 * address of the indirect rt frame , 0 otherwise
 */
static uint8_t find_short_rt_buffer(void *buf, void *short_addr)
{
    frame_info_t *data = (frame_info_t *)buf ;

    if (*(uint16_t *)short_addr ==
        convert_byte_array_to_16_bit(&data->mpdu[PL_POS_DST_ADDR_START]))
    {
        return 1;
    }
    return 0;
}
```

```
/*
 * @brief Handles transmission of pending CFP data
 *
 * This function handles the transmission of pending CFP data
 * frames in a beacon-enabled network, which need to be transmitted
 * immediately in the GTS I Time only.
 */
void mac_tx_pending_d_cfp_data_cb(void)
{
    frame_info_t *transmit_frame;
    retval_t tal_tx_status;

    /* This variable is used because we may put back a data packet
     * in the queue if the removed packet is not the right one, to avoid
     * Dead loop*/

    if(rt_indirect_data_q.size > 0)
    {
        /* search GTS data to be sent. */
        buffer_t *buf_ptr_next_data;
        search_t find_buf;
        /*
         * Look for pending data in the rt indirect queue for this short
         * address. */

        /*
         * Assign the function pointer for searching the
         * data having address of the requested device.
         */
        find_buf.criteria_func = find_short_rt_buffer;

        /* Update the short address to be searched. */
    }
```

```

        find_buf.handle = &d_gts_db[d_gts_counter-1].destAddr;

        /*
         * Read from the rt indirect queue. The removal of items from this
         * queue will be done after successful transmission of the frame.
         */
        buf_ptr_next_data = qmm_queue_read(&rt_indirect_data_q, &find_buf);
        /* Note: The find_buf structure is reused below, so do not change
this. */
        if (NULL == buf_ptr_next_data)
        {
            /* Nothing to be done. */
            return;
        }
        transmit_frame =
            (frame_info_t*)BMM_BUFFER_POINTER(buf_ptr_next_data);

        mac_trx_wakeup();

        transmit_frame->buffer_header = buf_ptr_next_data;
#if (MAC_INDIRECT_DATA_FFD == 1)
        /* Indirect transmission ongoing. */
        transmit_frame->indirect_in_transit = true;
#endif /* (MAC_INDIRECT_DATA_FFD == 1) */
        tal_tx_status= tal_tx_frame(transmit_frame, NO_CSMA_WITH_IFS,
true);
        if (MAC_SUCCESS == tal_tx_status)
        {
            MAKE_MAC_BUSY();
            qmm_queue_remove(&rt_indirect_data_q, &find_buf); // TAL will
take care of the packet removal
        }
        else
            transmit_frame->indirect_in_transit = false;
    }
}

```

```

/**
 *
 * @param callback_parameter
 */
void mac_t_d_cfp_usage_cb(void *callback_parameter)
{
    if(d_gts_counter < d_gts_count) // Still have GTS
    {
        d_gts_timerStep = GTS_RECV_ON;
        uint32_t now_time_symbols;
        pal_get_current_time(&now_time_symbols);
        now_time_symbols = TAL_CONVERT_US_TO_SYMBOLS(now_time_symbols);

        /* calculate absolute CFP start time (in symb) */
        uint32_t gts_stop_time;

#ifdef ACCURATE_D_GTS
        gts_stop_time = d_gts_db[d_gts_counter].length * aUnitBackoffPeriod;
#else
        gts_stop_time = GET_SLOT_DURATION(tal_pib.SuperFrameOrder);
        gts_stop_time *= d_gts_db[d_gts_counter].length;
#endif
    }
}

```

```

#endif /* ACCURATE_D_GTS */

    gts_stop_time          =          tal_add_time_symbols(now_time_symbols,
gts_stop_time);

    /* Set the current gts info */
    pal_timer_start(T_CFP_USAGE_Time,
                    TAL_CONVERT_SYMBOLS_TO_US(gts_stop_time),
                    TIMEOUT_ABSOLUTE,
                    (FUNC_PTR)mac_t_d_cfp_usage_cb,
                    NULL) ;
    d_gts_counter++;
    mac_tx_pending_d_cfp_data_cb();
}
else
{
    d_gts_timerStep = GTS_IDLE;
}
}

```

2. Dans les nœuds terminaux

```

/**
 * @brief Processes a received PRTPA frame
 *
 * This function will process a received GTS request command frame and
 * generates a MLME-GTS indication to the NHLE.
 *
 * @param GTS_req Specifies a pointer to the received GTS request frame
 */
void mac_process_prtpa_message(buffer_t *prtpa_frame)
{
#ifdef REDUCED_PARAM_CHECK
    /*
     * If the coordinator has macGTSPermit set to false, and receives a
     * GTS request command from a device, the command shall be ignored.
     */
    if (!mac_pib.mac_GTSPermit)
    {
        bmm_buffer_free(prtpa_frame);
        return;
    }
#endif /* REDUCED_PARAM_CHECK */
    /*
     * First GTS allocation/Deallocation
     */
    if (MAC_PAN_COORD_STARTED == mac_state) // I'm a PAN coord
    {
        bmm_buffer_free(prtpa_frame);
        return;
    }

    /* Stop PRTPA Wait Timer */
    pal_timer_stop(T_PRTPA_WAIT_Time);

    /*

```

```

    * Then, build the GTS indication.frame based on the frame reception
    buffer
    */
    mlme_prtpa_ind_t      *mpi              =      (mlme_prtpa_ind_t
*)BMM_BUFFER_POINTER(prtpa_frame);

    /* Build the MLME GTS indication parameters. */
    mpi->cmdcode = MLME_PRTPA_INDICATION ;
    mpi->d_gts_spec =
        mac_parse_data.mac_payload_data.prtpa_data.d_gts_spec;
    memcpy(mpi->d_gts_descriptors,
        mac_parse_data.mac_payload_data.prtpa_data.d_gts_descriptor_list,
        (3 * mpi->d_gts_spec ));

    /* Append the MLME associate indication to the MAC-NHLE queue. */
    qmm_queue_append(&mac_nhle_q, prtpa_frame) ;
}

```

```

/**
 *
 * @brief Enables the receiver to receive PRTPA frame
 *
 * This function will process a received GTS request command frame and
 * generates a MLME-GTS indication to the NHLE.
 */
void mac_prtpa_timer_cb(void)
{
    if (MAC_PRTPA_TIMER_SKIP_CFP == prtpa_timer_step)
    {
        /* Wake up node to receive PRTPA Pkt */
        uint8_t rx_enable_status = tal_rx_enable(PHY_RX_ON) ;

        if (RX_ON==rx_enable_status)
        {
            mac_rx_enabled = true;
        }
        prtpa_timer_step = MAC_PRTPA_TIMER_WAIT_PRTPA;
        uint32_t duration = GET_SLOT_DURATION(tal_pib.SuperFrameOrder);
        pal_timer_start(T_PRTPA_WAIT_Time,
            TAL_CONVERT_SYMBOLS_TO_US(duration),
            TIMEOUT_RELATIVE,
            (FUNC_PTR)mac_prtpa_timer_cb, NULL) ;
    }
    else if(MAC_PRTPA_TIMER_SKIP_D_CFP == prtpa_timer_step
        || MAC_PRTPA_TIMER_EMERGENCY_DELAY == prtpa_timer_step)
    {
        is_in_cap = 0x02;
        if(mac_is_pending_indirect)
        {
            if(mac_is_pending_indirect==1) // Short Addr pending
            {
                mac_build_and_tx_data_req(false, false, 0, NULL, 0);
            }
            else // ==2 , Long Addr pending
            {
                mac_build_and_tx_data_req(false, true, 0, NULL, 0);
            }
        }
    }
}

```

```

        else {
            mac_tx_pending_cap_data_cb();
        }
    }
}
/*
 * D-GTS recv usage timer callback
 */
void mac_t_d_gts_rx_usage_cb(void *callback_parameter)
{
    if (MAC_GTS_TIMER_DEFERING == d_gts_settings.timer_step)
    {
        uint32_t now_time_symbols;
        pal_get_current_time(&now_time_symbols);
        now_time_symbols = TAL_CONVERT_US_TO_SYMBOLS(now_time_symbols);

        /* calculate GTS duration */
        uint32_t d_gts_end_time;
#ifdef ACCURATE_D_GTS
        d_gts_end_time = d_gts_settings.length * aUnitBackoffPeriod;
#else
        d_gts_end_time = GET_SLOT_DURATION(tal_pib.SuperFrameOrder);
        d_gts_end_time *= d_gts_settings.length;
#endif /* ACCURATE_D_GTS */
        d_gts_end_time =
tal_add_time_symbols(now_time_symbols, d_gts_end_time);
        pal_timer_stop(T_GTS_RX_USAGE_Time); //stop if it's running
        retval_t status = pal_timer_start(T_GTS_RX_USAGE_Time,
TAL_CONVERT_SYMBOLS_TO_US(d_gts_end_time),
                                TIMEOUT_ABSOLUTE,
                                (FUNC_PTR)mac_t_d_gts_rx_usage_cb,
                                NULL);

        if (MAC_SUCCESS == status)
        {
            d_gts_settings.timer_step = MAC_GTS_TIMER_USING;
            uint8_t rx_enable_status = tal_rx_enable(PHY_RX_ON);
            if (RX_ON==rx_enable_status)
            {
                mac_rx_enabled = true;
            }
        }
    }
    else if (MAC_GTS_TIMER_USING == d_gts_settings.timer_step)
    {
        /* Set radio to sleep if allowed */
        mac_sleep_trans();
        d_gts_settings.timer_step = MAC_GTS_TIMER_IDLE;
    }
}

```

```

/*
 * @brief Handles transmission of pending CAP data
 *
 * This function handles the transmission of pending CAP data
 * frames in a beacon-enabled network, which need to be transmitted
 * immediately in the CAP I Time only.

```

```

*/
void mac_tx_pending_cap_data_cb(void)
{
    buffer_t *buf_ptr ;
    frame_info_t *transmit_frame ;
    retval_t tal_tx_status ;
    if(gts_data_q.size > 0)
    {
        buf_ptr = qmm_queue_read(&gts_data_q, NULL);
        if (NULL == buf_ptr)
        {
            /* Nothing to be done. */
            return;
        }

        transmit_frame = (frame_info_t *)BMM_BUFFER_POINTER(buf_ptr);
        mac_trx_wakeup();
        transmit_frame->buffer_header = buf_ptr;
        tal_tx_status= tal_tx_frame(transmit_frame, CSMA_SLOTTED, false);
        if (MAC_SUCCESS == tal_tx_status)
        {
            MAKE_MAC_BUSY();
            remove_frame_from_q(transmit_frame);
            //qmm_queue_remove(&gts_data_q, NULL); //remove from the front

            /* Return here, since for the rest of the packets this function
             * will be called back by mlme_gata_conf after the transmission
             * of the current packet
             */
            return;
        }
        else if (is_in_cap==0x02)
        {
            pal_timer_start(T_GTS_RX_USAGE_Time,
                           TAL_CONVERT_SYMBOLS_TO_US(MIN_TIMEOUT),
                           TIMEOUT_RELATIVE,
                           (FUNC_PTR)mac_tx_pending_cap_data_cb,
                           NULL);
        }
    }
}

/**
 * @brief Helper function to remove transmitted rt indirect data from the
 * queue
 *
 * @param f_ptr Pointer to frame_info_t structure of previously transmitted
 * frame
 */
static void remove_frame_from_q(frame_info_t *f_ptr)
{
    search_t find_buf;
    find_buf.criteria_func = find_buffer_cb;

    /* Update the address to be searched */
    find_buf.handle = (void *)f_ptr->buffer_header;
    qmm_queue_remove(&gts_data_q, &find_buf);
}

/**

```

```
* @brief Checks whether the rt indirect data frame address matches
* with the address passed.
*
* @param buf Pointer to indirect rt data buffer
* @param buffer Pointer to the buffer to be searched
*
* @return 1 if address matches, 0 otherwise */
static uint8_t find_buffer_cb(void *buf, void *buffer)
{
    uint8_t *buf_body = BMM_BUFFER_POINTER((buffer_t *)buffer);
    if (buf == buf_body)
        return 1 ;

    return 0 ;
}
```


Annexe C : Logiciel JSensorsMonitor

Nous avons développé l'application « JsensorMonitor » pour le recueil des données des capteurs et les paramètres réseau (LQI, RSSI), le suivi de l'évolution du réseau (Ajout/départ des nœuds), le calcul des métriques du réseau (c.-à-d. le délai). Ce logiciel est doté d'une interface graphique pour faciliter son utilisation. Il permet de visualiser la topologie du réseau, suivre l'évolution des grandeurs scalaires échantillonnées et de la qualité de transmission (en temps réel ou en temps différé), en plus d'autres fonctionnalités. Un exemple de ses services est illustré dans la figure 53. Ainsi nous avons à notre disposition un logiciel qui permet de traiter les données reçues du réseau réel sous plusieurs formats (numérique et graphique).

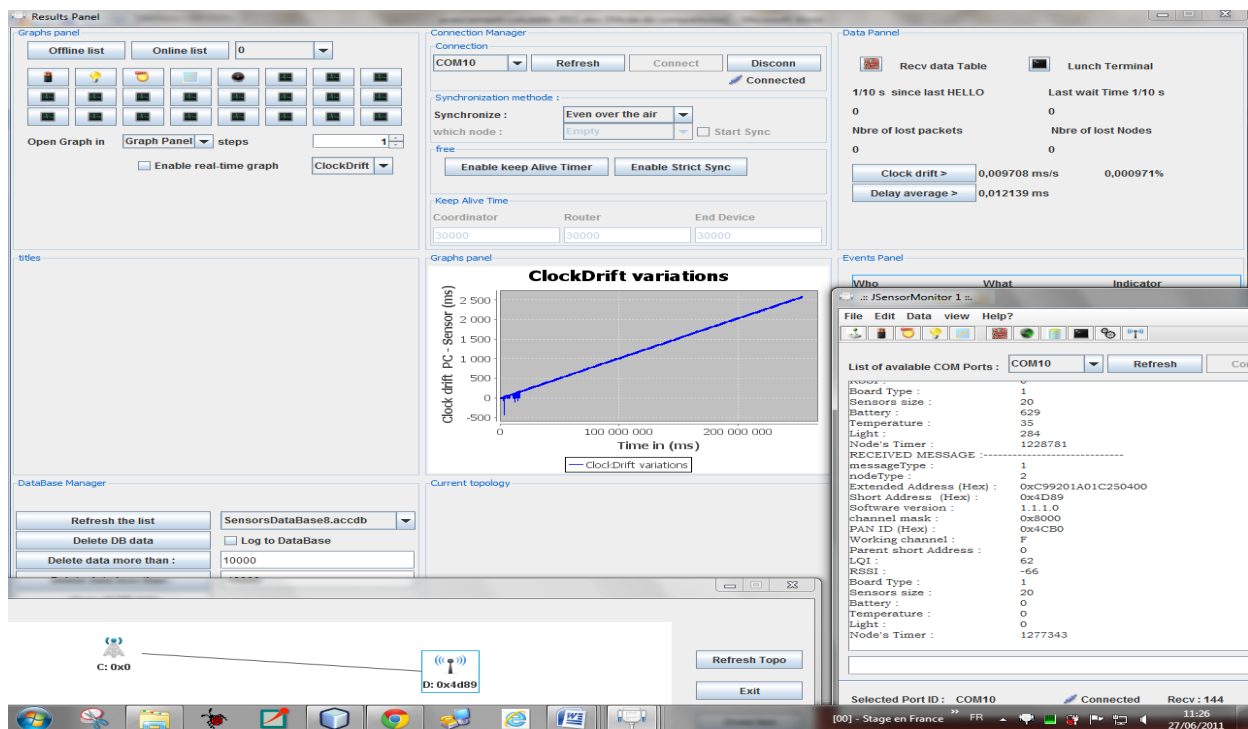


Figure 53 : Image de l'interface graphique du logiciel « JSensorsMonitor »

1. Architecture du logiciel

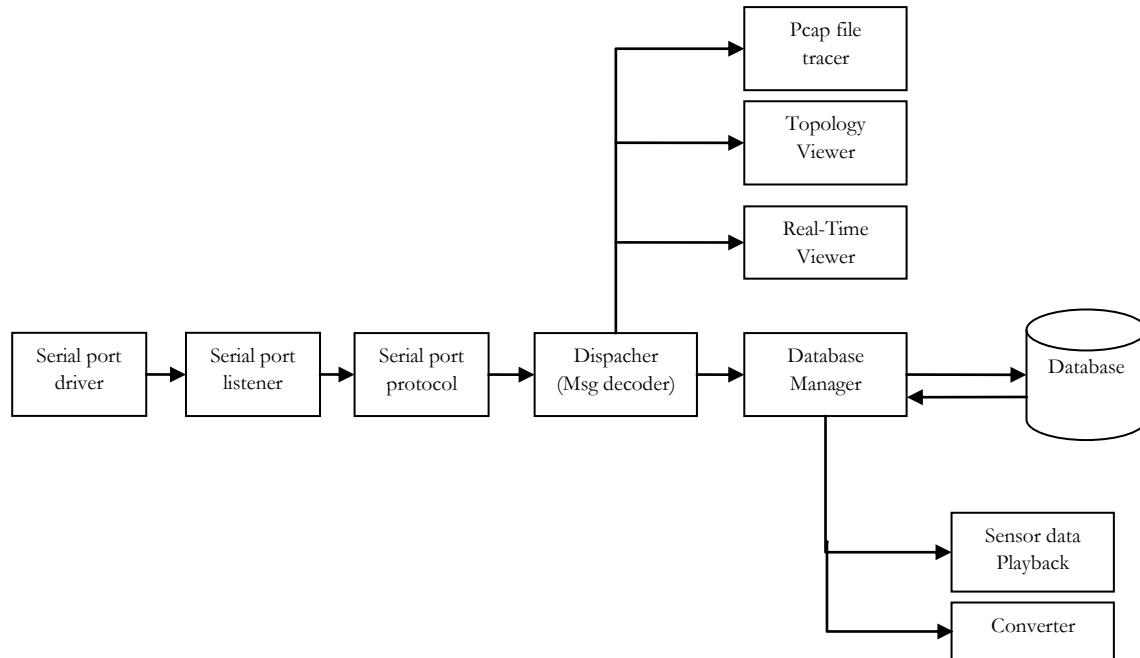


Figure 54 : Architecture globale du logiciel « JSensorsMonitor »

2. Fonctionnement du logiciel

Le nœud se connecte à l'ordinateur via le port série. Dès que les données soient reçues, par l'ordinateur, le pilote les remonte octet par octet à l'application connectée au port correspondant. La lecture est déclenché dans l'écouteur du port série qui constitue l'interface d'entrée du logiciel. Ce dernier utilise la bibliothèque externe RXTXcomm.jar avec le fichier dll « rxtxSerial.dll » [133] pour communiquer à travers le port RS232. Ensuite le logiciel applique le protocole de communication définit entre le nœud RCSF et le logiciel pour pouvoir identifier les messages dans le flux des octets transmis par les nœuds RCSF et vérifier leurs intégrités. Le code suivant représente le programme appliqué à chaque message transmis à travers l'interface UART des nœuds RCSF.

```
static uint8_t buildMessageToSendViaUart(void *data, void *newData,
uint8_t len_msdu)
{
    uint8_t *p, *q, 145um, I, cnt;
    q = data;
    p = newData;
    145um = 0;

    *p++ = 0x10;
    *p++ = 0x02 ;
    cnt = 2 ;

    for (I = 0; I < len_msdu; i++)
    {
        if (*q == 0x10)
        {
            *p++ = 0x10;
            145um += 0x10;
            cnt++;
        }
        145um += *q;
        *p++ = *q++;
        cnt++;
    }
    *p++ = 0x10;
    *p++ = 0x03;
    145um += 0x10 + 0x02 + 0x10 + 0x03;
    *p++ = 145um;
    cnt += 3 ;

    return cnt ;
}
```

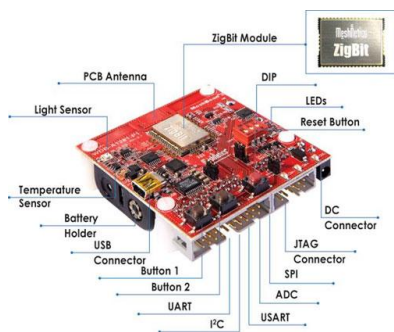
Du côté du logiciel JSensorsMonitor, le protocole exécute un programme permettant de extraire les messages à partir du flux d'octets transmis, vérifier son intégrité et de supprimer les octets introduit par le protocole (0x10 0x02, 0x10 0x03 et 0x10) en vue de récupérer le message originale.

Chaque message décodé est envoyé au « Dispatcher » qui l'envoie à son tour aux services correspondants selon la configuration de l'application et les interfaces actives (e.g. stockage dans la base de données, stockage dans fichier Pcap, visualisation en temps réel ...). La manipulation des fichier Pcap se fait grâce à la bibliothèque Open Source jNetPcap [134]. Alors que le dessin des courbes se fait à l'aide de la bibliothèque Open Source jFreeChart [135]

Les données stockées dans la base de données peuvent être consulté ultérieurement. Par l'application. Pour le format Pcap, on peut utiliser des outils d'analyse des paquets disponibles.

Annexe D : Illustrations des équipements utilisés dans les expériences

Les nœuds RCSF



Meshbean 2



STK600-Atmega128RFA1



Atmega128RFA1-RCB



iLive

Kits de développement



STK600



AVR JTAG ICE mk II Emulator

Liste des publications

1. **K. El Gholami**, K.-M. Hou, N. Elkamoun, H. Benabdillah, H. Shi, X. Liu « Experimental performance evaluation of FF-MAC protocol for real-time data transmission in WSN », *en cours*, 2014.
2. X. Liu, K. M. Hou, C. de Vault, H. Shi, et **K. El Gholami**, « MIROS: A Hybrid Real-Time Energy-Efficient Operating System for the Resource-Constrained Wireless Sensor Nodes », *Sensors*, vol. 14, no 9, p. 17621-17654, September 2014.
3. X. Liu, K. M. Hou, C. de Vault, et **K. El Gholami**, « Real-time Embedded Java Virtual Machine for Application Development in Wireless Sensor Network », *Journal of Networks*, vol. 9, no 7, ISSN 1796 2056, Juillet 2014.
4. **K. El Gholami**, N. Elkamoun, K.-M. Hou, Y. CHEN, J. CHANET, J. LI, « A new WPAN Model for NS-3 simulator », *International NICST'2013 Workshop, New and smart Information Communication Science and Technology to support Sustainable Development*, Clermont-Ferrand, France, 18-20 September 2013.
5. Yi-Bo Chen, K.-M. Hou, J Chanet, **K. El Gholami**, « A RPL based Adaptive and Scalable Data-collection Protocol module for NS-3 simulation platform », *International NICST'2013 Workshop, NICST'2013, New and smart Information Communication Science and Technology to support Sustainable Development*, Clermont-Ferrand, France, 18-20 September 2013.
6. **K. El Gholami**, N. Elkamoun et K.-M. Hou, « FF-MAC: Fast Forward IEEE 802.15.4 MAC Protocol for Real-Time Data Transmission », *International Journal of Advanced Computer Science and Applications*, vol. 4, n°. 1, p 25-32, Janvier 2013.
7. **K. El Gholami**, K.-M. Hou et N. Elkamoun, « Enhanced Superframe Structure of the IEEE802.15.4 Standard for Real-time Data Transmission in Star Network », *International Journal of Computer Applications*, vol. 51, n°. 15, p. 26-32, août 2012.

Bibliographie

- [1] N. Aakvaag, M. Mathiesen, et G. Thonet, « Timing and Power Issues in Wireless Sensor Networks - An Industrial Test Case », in *2012 41st International Conference on Parallel Processing Workshops*, Los Alamitos, CA, USA, 2005, p. 419-426.
- [2] S. Tilak, N. B. Abu-Ghazaleh, et W. Heinzelman, « A Taxonomy of Wireless Micro-Sensor Network Models », présenté à ACM MOBILE COMPUTING AND COMMUNICATIONS REVIEW, 2002.
- [3] Silicon Labs, « The Evolution of Wireless Sensor Networks ». [En ligne]. Disponible sur: <http://www.silabs.com/Support%20Documents/TechnicalDocs/evolution-of-wireless-sensor-networks.pdf>. [Consulté le: 05-juill-2014].
- [4] Q. Wang et I. Balasingham, « Wireless Sensor Networks - An Introduction », in *Wireless Sensor Networks: Application-Centric Design*, Y. K. Tan, Éd. InTech, 2010.
- [5] IEEE Computer Society, « Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) ». IEEE Std 802.15.4™-2006.
- [6] « ISA | The International Society of Automation ». [En ligne]. Disponible sur: <https://www.isa.org/>. [Consulté le: 04-août-2014].
- [7] « International Society of Automation News Release, ISA Standards & Practices Board ratifies ISA-100.11a document ». .
- [8] « ZigBee Alliance - Home ». [En ligne]. Disponible sur: <http://www.zigbee.org/>. [Consulté le: 04-août-2014].
- [9] « ZigBee Specifications ». [En ligne]. Disponible sur: <http://www.zigbee.org/Specifications.aspx>. [Consulté le: 04-août-2014].
- [10] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, et K. Pister, « System architecture directions for networked sensors », présenté à IN ARCHITECTURAL SUPPORT FOR PROGRAMMING LANGUAGES AND OPERATING SYSTEMS, 2000.
- [11] « Contiki: The Open Source Operating System for the Internet of Things ». [En ligne]. Disponible sur: <http://www.contiki-os.org/>. [Consulté le: 04-août-2014].
- [12] X. Liu, K. M. Hou, C. de Vault, H. Shi, et K. El Gholami, « MIROS: A Hybrid Real-Time Energy-Efficient Operating System for the Resource-Constrained Wireless Sensor Nodes », *Sensors*, vol. 14, n° 9, p. 17621-17654, sept. 2014.
- [13] L. Mainetti, L. Patrono, et A. Vilei, « Evolution of wireless sensor networks towards the Internet of Things: A survey », in *2011 19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2011, p. 1-6.
- [14] J. . Stankovic, « Research Directions for the Internet of Things », *IEEE Internet of Things Journal*, vol. 1, n° 1, p. 3-9, févr. 2014.
- [15] Z. Shelby et C. Bormann, *6LoWPAN: The Wireless Embedded Internet*. John Wiley & Sons, 2011.
- [16] « Technology Research | Gartner Inc. ». [En ligne]. Disponible sur: <http://www.gartner.com/technology/home.jsp>. [Consulté le: 13-nov-2014].
- [17] « Gartner Says 4.9 Billion Connected ». [En ligne]. Disponible sur: <http://www.gartner.com/newsroom/id/2905717>. [Consulté le: 13-nov-2014].
- [18] G. Xu, W. Shen, et X. Wang, « Applications of Wireless Sensor Networks in Marine Environment Monitoring: A Survey », *Sensors*, vol. 14, n° 9, p. 16932-16954, sept. 2014.

- [19] T. Dittrich, C. Menachem, Y. Herzel, et A. Lou, « Lithium batteries for wireless sensor networks ». Technical report, Tadiran Batteries, 2012.
- [20] G. Anastasi, M. Conti, M. Di Francesco, et A. Passarella, « Energy conservation in wireless sensor networks: A survey », *Ad Hoc Networks*, vol. 7, n° 3, p. 537-568, mai 2009.
- [21] V. Raghunathan, C. Schurgers, S. Park, et M. B. Srivastava, « Energy-aware wireless microsensor networks », *IEEE Signal Processing Magazine*, vol. 19, n° 2, p. 40-50, mars 2002.
- [22] Z. A. Eu, H.-P. Tan, et W. K. G. Seah, « Design and performance analysis of MAC schemes for Wireless Sensor Networks Powered by Ambient Energy Harvesting », *Ad Hoc Networks*, vol. 9, n° 3, p. 300-323, mai 2011.
- [23] A. Hande, T. Polk, W. Walker, et D. Bhatia, « Indoor solar energy harvesting for sensor network router nodes », *Microprocessors and Microsystems*, vol. 31, n° 6, p. 420-432, sept. 2007.
- [24] Atmel, « AVR2007: IEEE802.15.4 MAC power consumptions for AT86RF230 and ATmega1281 ». .
- [25] M. K. Stojcev, M. R. Kosanovic, et L. R. Golubovic, « Power management and energy harvesting techniques for wireless sensor nodes », in *9th International Conference on Telecommunication in Modern Satellite, Cable, and Broadcasting Services, 2009. TELSIKS '09*, 2009, p. 65-72.
- [26] G. E. Moore, « Cramming More Components Onto Integrated Circuits », *Proceedings of the IEEE*, vol. 86, n° 1, p. 82-85, janv. 1998.
- [27] K. Romer et F. Mattern, « The design space of wireless sensor networks », *IEEE Wireless Communications*, vol. 11, n° 6, p. 54-61, déc. 2004.
- [28] M. P. Durisic, Z. Tafa, G. Dimic, et V. Milutinovic, « A survey of military applications of wireless sensor networks », in *2012 Mediterranean Conference on Embedded Computing (MECO)*, 2012, p. 196-199.
- [29] K. Sohraby, D. Minoli, et T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*. John Wiley & Sons, 2007.
- [30] P. Kulakowski, E. Calle, et J. L. Marzo, « Performance study of wireless sensor and actuator networks in forest fire scenarios », *Int. J. Commun. Syst.*, vol. 26, n° 4, p. 515-529, avr. 2013.
- [31] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, et M. Welsh, « Deploying a wireless sensor network on an active volcano », *IEEE Internet Computing*, vol. 10, n° 2, p. 18-25, mars 2006.
- [32] A. Sieber, R. Karnapke, J. Nolte, et T. Martschei, « Using sensor technology to protect an endangered species: A case study », in *2011 IEEE 36th Conference on Local Computer Networks (LCN)*, 2011, p. 1044-1047.
- [33] D. C. Steere, A. Baptista, D. McNamee, C. Pu, et J. Walpole, « Research Challenges in Environmental Observation and Forecasting Systems », in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, New York, NY, USA, 2000, p. 292-299.
- [34] P. Corke, T. Wark, R. Jurdak, W. Hu, P. Valencia, et D. Moore, « Environmental Wireless Sensor Networks », *Proceedings of the IEEE*, vol. 98, n° 11, p. 1903-1917, nov. 2010.
- [35] H. Baldus, K. Klabunde, et G. Müsch, « Reliable Set-Up of Medical Body-Sensor Networks », in *Wireless Sensor Networks*, H. Karl, A. Wolisz, et A. Willig, Éd. Springer Berlin Heidelberg, 2004, p. 353-363.
- [36] H. Ding, H. Sun, et K. Hou, « Abnormal ECG signal detection based on compressed sampling in Wearable ECG sensor », in *2011 International Conference on Wireless Communications and Signal Processing (WCSP)*, 2011, p. 1-5.

- [37] M. Yusro, K. M. Hou, E. Pissaloux, H. L. Shi, K. Ramli, et D. Sudiana, « SEES: Concept and design of a smart environment explorer stick », in *2013 The 6th International Conference on Human System Interaction (HSI)*, 2013, p. 70-77.
- [38] F. Michahelles, P. Matter, A. Schmidt, et B. Schiele, « Applying wearable sensors to avalanche rescue », *Computers & Graphics*, vol. 27, n° 6, p. 839-847, déc. 2003.
- [39] C. Baouche, A. Freitas, et M. Misson, « Radio Proximity Detection in a WSN to Localize Mobile Entities within a Confined Area », *Journal of Communications*, vol. 4, n° 4, mai 2009.
- [40] J. Li et Y. Zhou, « Target Tracking in Wireless Sensor Networks », in *Wireless Sensor Networks: Application-Centric Design*, Y. K. Tan, Éd. InTech, 2010.
- [41] T. Dang et C. Devic, « OCARI: Optimization of communication for Ad hoc reliable industrial networks », in *6th IEEE International Conference on Industrial Informatics, 2008. INDIN 2008*, 2008, p. 688-693.
- [42] M. Erdelj, N. Mitton, et E. Natalizio, « Applications of Industrial Wireless Sensor Networks », *Industrial Wireless Sensor Networks: Applications, Protocols, and Standards*, mars 2013.
- [43] V. Ç. Güngör et G. P. Hancke, *Industrial Wireless Sensor Networks: Applications, Protocols, and Standards*. CRC Press, 2013.
- [44] S. Farahani, *ZigBee Wireless Networks and Transceivers*. Amsterdam; Boston: Newnes, 2008.
- [45] Y. Li et M. T. Thai, *Wireless Sensor Networks and Applications*. 2008.
- [46] J. Rabaey, J. Ammer, J., J. L. Da Silva, et D. Patel, « PicoRadio: Ad-hoc wireless networking of ubiquitous low-energy sensor/monitor nodes », in *IEEE Computer Society Workshop on VLSI, 2000. Proceedings*, 2000, p. 9-12.
- [47] C. Guo, L. C. Zhong, et J. M. Rabaey, « Low power distributed MAC for ad hoc sensor radio networks », in *IEEE Global Telecommunications Conference, 2001. GLOBECOM '01*, 2001, vol. 5, p. 2944-2948 vol.5.
- [48] E. Shih, P. Bahl, et M. J. Sinclair, « Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices », in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, New York, NY, USA, 2002, p. 160-171.
- [49] Y. Agarwal, R. Gupta, et C. Schurgers, « Dynamic power management using on demand paging for networked embedded systems », in *Design Automation Conference, 2005. Proceedings of the ASP-DAC 2005. Asia and South Pacific*, 2005, vol. 2, p. 755-759 Vol. 2.
- [50] C.-F. Chiasserini et R. R. Rao, « A distributed power management policy for wireless ad hoc networks », in *2000 IEEE Wireless Communications and Networking Conference, 2000. WCNC*, 2000, vol. 3, p. 1209-1213 vol.3.
- [51] IEEE Computer Society, « 802.11 IEEE Standard for Information technology ». 2012.
- [52] T. Park, K.-J. Park, et M. J. Lee, « Design and analysis of asynchronous wakeup for wireless sensor networks », *IEEE Transactions on Wireless Communications*, vol. 8, n° 11, p. 5530-5541, nov. 2009.
- [53] M. Riduan, E. Dutkiewicz, et X. Huang, « A Survey of Low Duty Cycle MAC Protocols in Wireless Sensor Networks », in *Emerging Communications for Wireless Sensor Networks*, A. Foerster et A. Foerster, Éd. InTech, 2011.
- [54] « ITU: International Telecommunication Union ». [En ligne]. Disponible sur: <http://www.itu.int>. [Consulté le: 05-août-2014].
- [55] M. Peuhkuri, « IP Quality of Service ». Helsinki University of Technology, Laboratory of Telecommunications Technology, 1999.
- [56] J. N. Al-Karaki et A. Kamal, « Routing techniques in wireless sensor networks: a survey », *IEEE Wireless Communications*, vol. 11, n° 6, p. 6-28, déc. 2004.

- [57] G. Lu, B. Krishnamachari, et C. S. Raghavendra, « An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks », *Wireless Communications and Mobile Computing*, vol. 7, n° 7, p. 863–875, sept. 2007.
- [58] G. Di Bacco, T. Melodia, et F. Cuomo, « A MAC protocol for delay-bounded applications in wireless sensor networks », *IN: PROC. MEDHOC-NET*, p. 208–220, 2004.
- [59] P. Patil et N. Patil, « Data Aggregation in Wireless Sensor Network », in *IEEE International Conference on Computational Intelligence and Computing Research, 2010*, 2010.
- [60] K. Dasgupta, K. Kalpakis, et P. Namjoshi, « An efficient clustering-based heuristic for data gathering and aggregation in sensor networks », in *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003*, 2003, vol. 3, p. 1948–1953 vol.3.
- [61] Z. Eskandari et F. Ayughi, « Data Aggregation Tree Construction: Algorithms and Challenges », in *Wireless Sensor Networks*, S. Tarannum, Éd. InTech, 2011.
- [62] E. Fasolo, M. Rossi, J. Widmer, et M. Zorzi, « In-network aggregation techniques for wireless sensor networks: a survey », *IEEE Wireless Communications*, vol. 14, n° 2, p. 70–87, avr. 2007.
- [63] W. Su et T. L. Lim, « Cross-Layer Design and Optimization for Wireless Sensor Networks », in *Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006*, 2006, p. 278–284.
- [64] « ISO/IEC 9646-7:1995 - Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework -- Part 7: Implementation Conformance Statements ». .
- [65] W. Ye, J. Heidemann, et D. Estrin, « An energy-efficient MAC protocol for wireless sensor networks », in *IEEE INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, 2002, vol. 3, p. 1567 - 1576 vol.3.
- [66] T. van Dam et K. Langendoen, « An adaptive energy-efficient MAC protocol for wireless sensor networks », in *Proceedings of the 1st international conference on Embedded networked sensor systems*, New York, NY, USA, 2003, p. 171–180.
- [67] S.-H. Yang, H.-W. Tseng, E. H.-K. Wu, et G.-H. Chen, « Utilization based duty cycle tuning MAC protocol for wireless sensor networks », in *IEEE Global Telecommunications Conference, 2005. GLOBECOM '05*, 2005, vol. 6, p. 5 pp. -3262.
- [68] T.-H. Hsu et J.-S. Wu, « An application-specific duty cycle adjustment MAC protocol for energy conserving over wireless sensor networks », *Computer Communications*, vol. 31, n° 17, p. 4081–4088, nov. 2008.
- [69] J. Mišić, S. Shafi, et V. B. Mišić, « The impact of MAC parameters on the performance of 802.15.4 PAN », *Ad Hoc Networks*, vol. 3, n° 5, p. 509–528, sept. 2005.
- [70] B. Gao, C. He, et L. Jiang, « Modeling and analysis of IEEE 802.15.4 CSMA/CA with sleep mode enabled », in *11th IEEE Singapore International Conference on Communication Systems, 2008. ICCS 2008*, 2008, p. 6–11.
- [71] Y.-K. Huang, A.-C. Pang, et H.-N. Hung, « A comprehensive analysis of low-power operation for beacon-enabled IEEE 802.15.4 wireless networks », *IEEE Transactions on Wireless Communications*, vol. 8, n° 11, p. 5601–5611, nov. 2009.
- [72] Z. Xiao, C. He, et L. Jiang, « Slot-based model for IEEE 802.15.4 MAC with sleep mechanism », *IEEE Communications Letters*, vol. 14, n° 2, p. 154–156, févr. 2010.
- [73] D. Dutta, A. Karmakar, et D. Kr. Saikia, « An Analytical Model for IEEE 802. 15. 4/ ZigBee Wireless Sensor Networks with Duty Cycle Mechanism for Performance Prediction and Configuration of MAC Parameters to Achieve QoS and Energy Efficiency », *International Journal of Computer Applications*, vol. 102, n° 5, p. 1–9, sept. 2014.

- [74] B. Nefzi et Y.-Q. Song, « QoS for wireless sensor networks: Enabling service differentiation at the MAC sub-layer using CoSenS », *Ad Hoc Networks*, vol. 10, n° 4, p. 680-695, juin 2012.
- [75] A. Koubaa, M. Alves, B. Nefzi, et Y.-Q. Song, « Improving the IEEE 802.15.4 Slotted CSMA/CA MAC for Time-Critical Events in Wireless Sensor Networks », présenté à Proceedings of the Workshop of Real-Time Networks (RTN 2006), Satellite Workshop to ECRTS 2006, 2006.
- [76] J.-G. Ko, Y.-H. Cho, et H. Kim, « Performance Evaluation of IEEE 802.15.4 MAC with Different Backoff Ranges in Wireless Sensor Networks », in *10th IEEE Singapore International Conference on Communication systems, 2006. ICCS 2006*, 2006, p. 1-5.
- [77] M. Kim et C.-H. Kang, « Priority-Based Service-Differentiation Scheme for IEEE 802.15.4 Sensor Networks in Nonsaturation Environments », *IEEE Transactions on Vehicular Technology*, vol. 59, n° 7, p. 3524-3535, sept. 2010.
- [78] T. H. Kim et S. Choi, « Priority-based delay mitigation for event-monitoring IEEE 802.15.4 LR-WPANs », *IEEE Communications Letters*, vol. 10, n° 3, p. 213-215, mars 2006.
- [79] V. P. Rao et D. Marandin, « Adaptive Backoff Exponent Algorithm for Zigbee (IEEE 802.15.4) », in *Proceedings of the 6th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking*, Berlin, Heidelberg, 2006, p. 501-516.
- [80] Y.-G. Hong, H.-J. Kim, H.-D. Park, et D.-H. Kim, « Adaptive GTS allocation scheme to support QoS and multiple devices in 802.15.4 », in *Proceedings of the 11th international conference on Advanced Communication Technology - Volume 3*, Piscataway, NJ, USA, 2009, p. 1697-1702.
- [81] L. Cheng, X. Zhang, et A. G. Bourgeois, « GTS allocation scheme revisited », *Electronics Letters*, vol. 43, n° 18, p. 1005 -1006, 2007.
- [82] Y. Zhou, Y. Wang, J. Ma, J. Jia, et F. Wang, « A Low-Latency GTS Strategy in IEEE802.15.4 for Industrial Applications », in *Second International Conference on Future Generation Communication and Networking, 2008. FGCN '08*, 2008, vol. 1, p. 411 -414.
- [83] C. Na, Y. Yang, et A. Mishra, « An optimal GTS scheduling algorithm for time-sensitive transactions in IEEE 802.15.4 networks », *Computer Networks*, vol. 52, n° 13, p. 2543-2557, sept. 2008.
- [84] A. van den Bossche, T. Val, et E. Campo, « Modelisation and validation of a full deterministic medium access method for IEEE 802.15.4 WPAN », *Ad Hoc Networks*, vol. 7, n° 7, p. 1285-1301, sept. 2009.
- [85] Y.-K. Huang, A.-C. Pang, et H.-N. Hung, « An Adaptive GTS Allocation Scheme for IEEE 802.15.4 », *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, n° 5, p. 641-651, mai 2008.
- [86] X. J. Dong, « Adaptive polling algorithm for PCF mode of IEEE 802.11 wireless LANs », *Electronics Letters*, vol. 40, n° 8, p. 482, 2004.
- [87] B. Shrestha, E. Hossain, S. Camorlinga, R. Krishnamoorthy, et D. Niyato, « An Optimization-Based GTS Allocation Scheme for IEEE 802.15.4 MAC with Application to Wireless Body-Area Sensor Networks », in *2010 IEEE International Conference on Communications (ICC)*, 2010, p. 1-6.
- [88] M. . Hanson, H. C. Powell, A. Barth, K. Ringgenberg, B. H. Calhoun, J. H. Aylor, et J. Lach, « Body Area Sensor Networks: Challenges and Opportunities », *Computer*, vol. 42, n° 1, p. 58-65, janv. 2009.
- [89] A. Koubaa, M. Alves, E. Tovar, et A. Cunha, « An implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks: theory and practice », *Real-Time Syst.*, vol. 39, n° 1-3, p. 169-204, août 2008.

- [90] A. Koubaa, M. Alves, et E. Tovar, « A comprehensive simulation study of slotted CSMA/CA for IEEE 802.15.4 wireless sensor networks », in *2006 IEEE International Workshop on Factory Communication Systems*, 2006, p. 183-192.
- [91] Y.-K. Huang, A.-C. Pang, et T.-W. Kuo, « AGA: Adaptive GTS Allocation with Low Latency and Fairness Considerations for IEEE 802.15.4 », in *IEEE International Conference on Communications, 2006. ICC '06*, 2006, vol. 9, p. 3929-3934.
- [92] F. Xia, A. Vinel, R. Gao, L. Wang, et T. Qiu, « Evaluating IEEE 802.15.4 for Cyber-Physical Systems », *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, n° 1, p. 596397, mars 2011.
- [93] J. Jeon, J. W. Lee, H. S. Kim, et W. H. Kwon, « PECAP: Priority-Based Delay Alleviation Algorithm for IEEE 802.15.4 Beacon-Enabled Networks », *Wireless Pers Commun*, vol. 43, n° 4, p. 1625-1631, déc. 2007.
- [94] P. Di Marco, P. Park, C. Fischione, et K. H. Johansson, « TRENd: A Timely, Reliable, Energy-Efficient and Dynamic WSN Protocol for Control Applications », in *2010 IEEE International Conference on Communications (ICC)*, 2010, p. 1-6.
- [95] M. Di Francesco, G. Anastasi, M. Conti, S. K. Das, et V. Neri, « Reliability and Energy-Efficiency in IEEE 802.15.4/ZigBee Sensor Networks: An Adaptive and Cross-Layer Approach », *IEEE Journal on Selected Areas in Communications*, vol. 29, n° 8, p. 1508-1524, sept. 2011.
- [96] D. Kliazovich et F. Granelli, « Introduction: Why Cross-Layer? Its Advantages and Disadvantages », p. 1 - 32, 2010.
- [97] IEEE Computer Society, « Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), Amendment 1: MAC sublayer ». IEEE Std 802.15.4eTM-2012.
- [98] IEEE Computer Society, « Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) ». IEEE Std 802.15.4TM-2011.
- [99] D. Abbott, *Linux for Embedded and Real-time Applications*. Newnes, 2012.
- [100] K. El Gholami, K.-M. Hou, et N. Elkamoun, « Enhanced Superframe Structure of the IEEE802.15.4 Standard for Real-time Data Transmission in Star Network », *International Journal of Computer Applications*, vol. 51, n° 15, p. 26-32, août 2012.
- [101] K. El Gholami, N. Elkamoun, et K.-M. Hou, « FF-MAC: Fast Forward IEEE 802.15.4 MAC Protocol for Real-Time Data Transmission », *International Journal of Advanced Computer Science and Applications*, vol. 4, n° 1, p. 25-32, janv. 2013.
- [102] S. Fan, J. Li, H. Sun, et R. Wang, « Throughput analysis of GTS allocation in beacon enabled IEEE 802.15.4 », in *2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, 2010, vol. 4, p. 561-565.
- [103] « The Network Simulator - ns-2 ». [En ligne]. Disponible sur: <http://www.isi.edu/nsnam/ns/>. [Consulté le: 24-mars-2014].
- [104] T. Issariyakul et E. Hossain, *Introduction to Network Simulator NS2*, 1^{re} éd. Springer Publishing Company, Incorporated, 2008.
- [105] « Contributed Code - Nsnam ». [En ligne]. Disponible sur: http://nsnam.isi.edu/nsnam/index.php/Contributed_Code. [Consulté le: 24-mars-2014].
- [106] J. Zheng, Myung J. Lee, « A comprehensive performance study of IEEE 802.15.4 », *Sensor Network Operations*, IEEE Press, Wiley Interscience, vol. 4, p. 218-237, 2006.
- [107] Y. Chen, J. P. Chanet, K. M. Hou, et P. Zhou, « A context-aware tool-set for routing-targeted mutual configuration and optimization of LLNs through bridging virtual and physical worlds », in *NICST'2014, New and smart Information Communication Science and Technology to support Sustainable Development*, Weihai, Chine, 2014, p. 5 p.

- [108] A. Rachedi, S. Lohier, S. Cherrier, et I. Salhi, « Wireless Network Simulators Relevance Compared to a Real Testbed in Outdoor and Indoor Environments », in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, New York, NY, USA, 2010, p. 346–350.
- [109] R. E. Shannon, « Introduction to the Art and Science of Simulation », in *Proceedings of the 30th Conference on Winter Simulation*, Los Alamitos, CA, USA, 1998, p. 7–14.
- [110] « Two-ray ground reflection model ». [En ligne]. Disponible sur: <http://www.isi.edu/nsnam/ns/doc/node218.html>. [Consulté le: 27-oct-2014].
- [111] « NetAnim - Nsnam ». [En ligne]. Disponible sur: <http://www.nsnam.org/wiki/index.php/NetAnim>. [Consulté le: 24-mars-2014].
- [112] K. El Gholami, K.-M. Hou, N. Elkamoun, H. Benabdillah, H. L. Sha, et X. Liu, « Experimental performance evaluation of FF-MAC protocol for real-time data transmission in WSN », *submitted*, juill. 2014.
- [113] ATMEL, « Atmel AVR2025: IEEE 802.15.4 MAC Software Package - User Guide ». <http://www.atmel.com/Images/doc8412.pdf>.
- [114] « Omnet++, Discrete Event Simulation System ». [En ligne]. Disponible sur: <http://www.omnetpp.org/>. [Consulté le: 04-juill-2014].
- [115] « SteelCentral for Performance Management and Control | Riverbed application and network performance management solutions | Riverbed ». [En ligne]. Disponible sur: <http://www.riverbed.com/products/performance-management-control/opnet.html>. [Consulté le: 04-juill-2014].
- [116] P. Jurcik, A. Koubaa, M. Alves, E. Tovar, et Z. Hanzalek, « A Simulation Model for the IEEE 802.15.4 protocol: Delay/Throughput Evaluation of the GTS Mechanism », in *15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007. MASCOTS '07*, 2007, p. 109–116.
- [117] « Castalia, Wireless Sensor Network Simulator ». [En ligne]. Disponible sur: <https://castalia.forge.nicta.com.au/index.php/en/>. [Consulté le: 04-juill-2014].
- [118] « MiXiM project ». [En ligne]. Disponible sur: <http://mixim.sourceforge.net/>. [Consulté le: 04-juill-2014].
- [119] F. Chen et F. Dressler, « A Simulation Model of IEEE 802.15.4 in OMNeT », in *In 6. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze, Poster Session*, 2007, p. 35–38.
- [120] E. Weingartner, H. vom Lehn, et K. Wehrle, « A Performance Comparison of Recent Network Simulators », in *IEEE International Conference on Communications, 2009. ICC '09*, 2009, p. 1–5.
- [121] IEEE Computer Society, « 802.15.4 IEEE Standard for Information technology ». 2006.
- [122] H. Wu, S. Nabar, et R. Poovendran, « An energy framework for the network simulator 3 (NS-3) », in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, ICST, Brussels, Belgium, Belgium, 2011, p. 222–230.
- [123] G. Montenegro, N. Kushalnagar, J. Hui, et D. Culler, « RFC 4944-IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN) RFC ». .
- [124] K. El Gholami, N. Elkamoun, K. M. Hou, C. Yibo, J.-P. chanet, et J.-J. Li, « New WPAN Model for NS-3 simulator », présenté à New and smart Information Communication Science and Technology to support Sustainable Development, 2013.
- [125] Pecorella et Tommaso, « ns-3-6LoWPAN ». [En ligne]. Disponible sur: <http://code.nsnam.org/tpecorella/ns-3-6LoWPAN/>. [Consulté le: 29-mai-2013].
- [126] M. Stoffers et G. Riley, « Comparing the ns-3 Propagation Models », in *2012 IEEE 20th International Symposium on Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2012, p. 61–67.

- [127] C. Yi-bo, K. M. Hou, J.-P. chanet, et K. El Gholami, « A RPL based Adaptive and Scalable Data-collection Protocol module for NS-3 simulation platform », présenté à New and smart Information Communication Science and Technology to support Sustainable Development, 2013.
- [128] T. W. <wintert@acm.org>, « RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks ». [En ligne]. Disponible sur: <https://tools.ietf.org/html/rfc6550>. [Consulté le: 06-nov-2014].
- [129] S. Chakrabarti, Z. Shelby, et E. Nordmark, « Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) ». [En ligne]. Disponible sur: <http://tools.ietf.org/html/rfc6775>. [Consulté le: 06-nov-2014].
- [130] Z. Shelby, K. Hartke, et C. Bormann, « The Constrained Application Protocol (CoAP) ». [En ligne]. Disponible sur: <https://tools.ietf.org/html/rfc7252>. [Consulté le: 06-nov-2014].
- [131] H.-L. Shi, K. M. Hou, H.-Y. Zhou, et X. Liu, « Energy Efficient and Fault Tolerant Multicore Wireless Sensor Network: E²MWSN », in *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, 2011, p. 1 -4.
- [132] « ILive Platform ». [En ligne]. Disponible sur: <http://edss.isima.fr/auth/home.php>. [Consulté le: 29-mai-2013].
- [133] « RXTX: The Prescription for Transmission ». [En ligne]. Disponible sur: <http://users.frii.com/jarvi/rxtx/index.html>. [Consulté le: 03-nov-2014].
- [134] « jNetPcap OpenSource ». [En ligne]. Disponible sur: <http://jnetpcap.com/?q=node>. [Consulté le: 03-nov-2014].
- [135] « JFreeChart ». [En ligne]. Disponible sur: <http://www.jfree.org/jfreechart/>. [Consulté le: 03-nov-2014].

Glossaire

Les abréviations et les acronymes sont organisés dans un ordre alphabétique :

A

ACK	ACKnowledgement
AIMD	Additive-Increase/Multiplicative-Decrease
API	Application Programming Interface
ASK	Amplitude Shift Keying

B

BE	Backoff Exponent
Ben	Beacon-Enabled
BI	Beacon Interval
BMM	Buffer ManageMent
BO	Beacon Order
BPSK	Binary Phase-Shift Keying

C

CAP	Collision Access Period
CCA	Clear Channel Assessment
CFP	Collision Free Period
CoAP	Constrained Application Protocol
CRC	Cyclic Redundancy Check
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear To Send
CWD	Contention Window Differentiation

D

D-CFP	Dynamic – Collision Free Period
D-GTS	Dynamically allocated Guaranteed Time Slot
DC	Duty Cycle
DSME	Deterministic and Synchronous Multi-channel Extension

E

ED	Energy Detection
-----------	------------------

F

FF-MAC	Fast-Forward MAC
FFD	Full Function Device
FIFO	First In First Out
FRT	Frame Tailoring

G

GPIO	General Purpose Input/Output
GTS	Guaranteed Time Slot

I

IEEE	Institute of Electrical and Electronics Engineers
IFS	Inter-Frame Spacing
IoT	Internet of Things
ISA	International Society of Automation
ISM	Industrial, Scientific and Medical
ISO	International Organization for Standardization
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector

L

LIFS	Long Inter-Frame Spacing
LL-Beacon	Low Latency Beacon
LLDN	Low Latency Deterministic Network
LQI	Link Quality Indicator
LR-WPAN	Low Rate Wireless Personal Area Network

M

MAC	Medium Access Control
MCL	MAC Core Layer
MCPS	MAC Common Part Sublayer
MCU	Micro-Controller Unit
MEMS	Micro-Electro-Mechanical System
MLME	MAC Layer Management Entity
MPIB	MAC PAN Information Base

N

NBEn	Non Beacon-Enabled
ND	Neighbor discovery

O

OMNeT++	Objective Modular Network Test-bed in C++
OPNET	Optimized Network Engineering Tools
O-QPSK	Offset – Quadrature Shift Keying
OSI	Open System Interconnection

P

PAL	Platform Abstraction Layer
------------	----------------------------

PAN	Personal Area Network
PCAP	Packet CAPture
PD	PHY Data
PDR	Packet Delivery Ratio
PHY	PHYsical layer
PLME	PHY Layer Management Entity
POS	Personal Operating Space
PPIB	PHY PAN Information Base
PRTPA	Pending Real-Time Packets Advertisement

Q

QMM	Queue ManagMent
QoS	Quality of Service

R

RCSF	Réseaux de Capteurs Sans Fil
RFD	Reduced Function Device
RPL	Routing Protocol for Low power and Lossy networks
RSSI	Received Signal Strength Indication
RTS	Request To Send

S

SAL	Security Abstraction Layer
SAP	Service Access Point
SD	Superframe Duration
sd	slot duration
SIFS	Short Inter-Frame Spacing
SNR	Signal to Noise Ratio

SO Superframe Order

SOSUS Sound Surveillance System

SSCS Service-Specific Convergence Sub-layer

STB Security ToolBox

T

TAL Transceiver Abstraction Layer

TDMA Time Division Multiple Access

TFA Transceiver Feature Access

TREnD Timely, Reliable, Energy-efficient and Dynamic

U

UDP User Datagram Protocol

V

VQ Virtual Queue

W

Wi-Fi Wireless Fidelity

WoT Web of Things

WPAN Wireless Personal Area Network

Table des illustrations

Figures :

FIGURE 1 : GAIN DU MARCHÉ DES RCSFs AVEC LA BAISSÉ DES COUTS DES CAPTEURS [3]	15
FIGURE 2 : TAXONOMIE DES ALGORITHMES DE TRAÇABILITÉ DANS LES RCSF [40]	20
FIGURE 3 : EXEMPLE DE RÉSEAU DE CAPTEURS DANS UN BATIMENT RESIDENTIEL TYPIQUE [44]	21
FIGURE 4 : STRUCTURE DE LA SUPERTRAME AVEC LES PERIODES OPTIONNELLES	29
FIGURE 5 : ARCHITECTURE D'UN NŒUD LR-WPAN [5]	31
FIGURE 6 : ÉCART ENTRE L'ENVOI DU 1 ^{ER} PAQUET DANS LA CAP ET LE PREMIER PAQUET DANS LA CFP	47
FIGURE 7 : NOUVEL ORDRE DES DEUX PERIODES CAP ET CFP DANS LA NOUVELLE SUPERTRAME	50
FIGURE 8 : EXEMPLE DE LA NOUVELLE STRUCTURE DE LA SUPERTRAME SELON LE PROTOCOLE FF-MAC	52
FIGURE 9 : DIAGRAMME DE SEQUENCE DECRIVANT LE CADENCEMENT DES ECHANGES PENDANT LA SECTION ACTIVE DE LA SUPERTRAME	53
FIGURE 10 : FORMAT DE LA TRAME PRTPA	54
FIGURE 11 : VALEURS DES BITS DU CHAMP DE CONTROLE DE LA TRAME PRTPA	54
FIGURE 12 : FORMAT D'UN DESCRIPTEUR D-GTS	55
FIGURE 13 : FORMAT DU CHAMP DE SPECIFICATION DE LA SUPERTRAME DEFINI DANS LE STANDARD IEEE 802.15.4 ET LA MODIFICATION APPORTEE PAR LE PROTOCOLE FF-MAC	56
FIGURE 14 : ALGORITHME SUIVI PAR UN NŒUD TERMINAL LORS DE LA RECEPTION DE LA TRAME PRTPA	59
FIGURE 15 : EXEMPLE D'UNE TOPOLOGIE DE CAPTEURS/ACTUATEURS	61
FIGURE 16 : EXEMPLE D'ÉCHANGE DES PAQUETS TEMPS-REEL SELON LE PROTOCOLE FF-MAC	62
FIGURE 17 : ARCHITECTURE DE BASE DE NS-2 (SOURCE [104])	67
FIGURE 18 : EXEMPLE D'UN RÉSEAU DE 51 NŒUDS VISUALISÉ À L'AIDE DE NAM [111]	70
FIGURE 19 : DELAI DE BOUT EN BOUT EN FONCTION DU PARAMÈTRE BO (21 NŒUDS, SO=5)	71
FIGURE 20 : DELAI DE BOUT-EN-BOUT EN FONCTION DU BO « BEACON ORDER », AVEC (N=31 ET SO=4)	72
FIGURE 21 : DELAI DE BOUT-EN-BOUT EN FONCTION DU RAPPORT CYCLIQUE POUR UN RÉSEAU DE 21 NŒUDS	73
FIGURE 22 : DELAI DE BOUT EN BOUT POUR DIFFÉRENTES DENSITÉS DE RÉSEAU	73
FIGURE 23 : DELAI DE BOUT-EN-BOUT EN FONCTION DU PARAMÈTRE BO POUR DIFFÉRENTES DENSITÉS DES NŒUDS	74
FIGURE 24 : RÉSUMÉ DES SCÉNARIOS DE SIMULATION DU PROTOCOLE FF-MAC	75
FIGURE 25 : MATÉRIEL UTILISÉ DANS LES EXPÉRIENCES	77
FIGURE 26 : ARCHITECTURE DU LOGICIEL MAC D'ATMEL POUR LES NŒUDS IEEE 802.15.4 [113]	79
FIGURE 27 : INTERFACE PRINCIPALE DU LOGICIEL « JSensorsMonitor »	81
FIGURE 28 : SCHEMA DE MESURES	83
FIGURE 29 : TRANSMISSION DES TRAMES SELON LE PROTOCOLE FF-MAC	85
FIGURE 30 : PARTIE DES ECHANGES ENREGISTRES PAR NOTRE « SNIFFER » ET AFFICHEE SUR LE LOGICIEL « WIRESHARK »	87
FIGURE 31 : DELAI DE COMMUNICATION ENTRE LE COORDINATEUR ET UN NŒUD DU RÉSEAU PROPRIÉTAIRE D'UN GTS	88
FIGURE 32 : DELAI DE BOUT EN BOUT POUR CHAQUE PAQUET (BO=7, SO=4)	90
FIGURE 33 : DELAI DE BOUT EN BOUT EN FONCTION DU PARAMÈTRE SO (BO = SO)	91
FIGURE 34 : DELAI DE BOUT-EN-BOUT EN FONCTION DU PARAMÈTRE BO (SO=4)	92
FIGURE 35 : DELAI DE BOUT-EN-BOUT EN FONCTION DU RAPPORT CYCLIQUE (SO=4)	93
FIGURE 36 : DELAI DE BOUT-EN-BOUT EN FONCTION DE LA DENSITÉ DES NŒUDS	94
FIGURE 37 : COMPARAISON DES PERFORMANCES DES NŒUDS UTILISANT LA PERIODE CAP (CSMA-CA) DANS LE CAS DU PROTOCOLE FF-MAC ET LE STANDARD IEEE 802.15.4 (DC = 100%)	95

FIGURE 38 : DEBITS MESURES EN FONCTION DES NOMBRES DE NŒUDS (DC=12.5%)	96
FIGURE 39 : ARCHITECTURE DU MODELE WPAN	103
FIGURE 40 : ARCHITECTURE INTERNE DU MODULE WPANMAC	105
FIGURE 41 : MODELE D'ENERGIE D'UN NŒUD WPAN 'WPAN ENERGY DEVICE MODEL'	108
FIGURE 42 : EXEMPLE D'UTILISATION DE WIRESHARK POUR VISUALISER LE PAQUET UDP PORTE PAR DES TRAMES IEEE 802.15.4	110
FIGURE 43 : EXEMPLE DE RESEAU WPAN VISUALISE A L'AIDE DE L'ANIMATEUR « NETANIM »	111
FIGURE 44 : PUISSANCE REÇUE EN FONCTION DE LA DISTANCE « SOURCE-DESTINATION » SANS INTERFERENCES (2 NŒUDS, MODELE DE PROPAGATION FRIIS)	115
FIGURE 45 : LQI EN FONCTION DE LA DISTANCE « SOURCE-DESTINATION » SANS INTERFERENCES (2 NŒUDS, MODELE DE PROPAGATION FRIIS)	115
FIGURE 46 : PUISSANCE REÇUE EN FONCTION DE LA DISTANCE « SOURCE-DESTINATION » EN PRESENCE DES INTERFERENCES (20 NŒUDS, MODELE DE PROPAGATION FRIIS)	116
FIGURE 47 : LQI EN FONCTION DE LA DISTANCE « SOURCE-DESTINATION » EN PRESENCE DES INTERFERENCES (20 NŒUDS, MODELE DE PROPAGATION FRIIS)	117
FIGURE 48 : DELAI EN FONCTION DU NOMBRE POUR LES DIFFERENTES COUCHES PHY	119
FIGURE 49 : DELAI EN FONCTION DU « DUTY CYCLE » POUR DIFFERENTES DENSITES	120
FIGURE 50 : DEBIT UTILE EN FONCTION DES TAUX D'ENVOI DES PAQUETS UDP	121
FIGURE 51 : DEBITS UTILES MESURES POUR LES DIFFERENTES COUCHES PHYSIQUES	122
FIGURE 52 : DEBITS BINAIRES MESURES EN UTILISANT LES DIFFERENTES COUCHES PHYSIQUES DU STANDARD IEEE 802.15.4 POUR DES NOMBRES DE NŒUDS DIFFERENTS	123
FIGURE 53 : IMAGE DE L'INTERFACE GRAPHIQUE DU LOGICIEL « JSensorsMonitor »	143
FIGURE 54 : ARCHITECTURE GLOBALE DU LOGICIEL « JSensorsMonitor »	144

Tableaux :

TABEAU 1 : MODES DE FONCTIONNEMENT DU MODULE RADIO AT86RF230	16
TABEAU 2 : BANDES DE FREQUENCE ET DEBITS BINAIRES ET DE SYMBOLES DISPONIBLES [5]	28
TABEAU 3 : VALEURS DU CHAMP « TYPE DE TRAME »	53
TABEAU 4 : VALEURS DE QUELQUES CONSTANTES MAC POUR LE CALCUL DES IFS	58
TABEAU 5 : RESUME DES PARAMETRES DE SIMULATION.....	70
TABEAU 6 : CARACTERISTIQUES DES PLATEFORMES UTILISEES DANS L'EVALUATION DE PERFORMANCES	78
TABEAU 7 : CANAUX PROPRIETAIRES D'ATMEL POUR LES HAUTS DEBITS BINAIRES NON-STANDARD	78
TABEAU 8 : UTILISATION DE MEMOIRE PAR L'IMPLEMENTATION DU PROTOCOLE FF-MAC ET CELLE DU STANDARD IEEE 802.15.4	84
TABEAU 9 : PARAMETRES COMMUNS DE SIMULATION.....	112

La Gestion de la Qualité de Service Temps-Réel dans les Réseaux de Capteurs Sans Fil

Résumé:

Le monde des réseaux de capteurs sans fil 'RCSF' a connu de grands progrès au cours de ces dernières années. Ainsi, les RCSF ont pu intégrer divers champs d'application (environnement, militaire, médecine, domotique ...) dont quelques-uns ayant des exigences en termes de qualité de service 'QoS'. Cependant, la garantie de la QoS dans un RCSF pose des problèmes de recherche non triviaux, à cause de la nature peu fiable de la communication sans fil et des limitations des ressources des nœuds RCSF (processeur, mémoire, énergie ...). Dans le cadre de cette thèse, nous nous sommes focalisés sur les protocoles MAC afin d'améliorer le support de QoS dans les RCSF. Plus particulièrement, le standard IEEE 802.15.4. Les mécanismes de conservation d'énergie proposés par ce standard sont efficaces et présentent une flexibilité par rapport aux besoins des applications. Néanmoins, le temps de sommeil des nœuds affecte considérablement le délai de communication, donnant naissance à un compromis énergie/délai. Pour résoudre ce problème, nous avons amélioré ce standard pour un meilleur support des applications temps-réel. L'approche proposée permet de réduire le délai de communication de manière significative, même pour de faibles rapports cycliques, grâce au nouveau format de la supertrame. Les performances ont été validées par simulation et sur des plateformes de nœuds RCSF réelles. Nous avons proposé également un modèle de simulation pour le standard IEEE 802.15.4 sous le simulateur NS-3.

Mots-clefs : Réseaux de capteurs sans fil; Qualité de Service; Temps-réel ; Rapport cyclique ; FF-MAC; IEEE 802.15.4; NS-3; Prototypage

Real-Time Quality of Service Management in Wireless Sensor Networks

Abstract:

In the last years, Wireless Sensor Networks ‘WSN’ knew a tremendous evolution which attracted many applications. WSN has several characteristics that make it a unique research field, such as, WSN nodes’ constraints and the unreliable (lossy) wireless communication. The IEEE 802.15.4 standard is the first standard designed for this type of networks known as LR-WPANs ‘Low-Rate Wireless Personal Area Network’. The energy conservation mechanism proposed by the current standard is quite efficient and very flexible. This flexibility comes from the ability to configure different duty cycles to meet specific applications’ requirements. However, this mechanism has a considerable impact on the end-to-end delay. Our approach resolves the energy/delay trade-off by avoiding the storage of the real-time data in the coordinator during sleep time, more particularly in Multi-source Multi-sink networks. A new superframe structure is adopted and a deterministic reception scheduling is used. In this thesis, we also proposed a new WPAN model for the Network Simulator 3 ‘NS-3’.

Keywords: Wireless Sensor Networks; Quality of Service; Real-Time; Duty Cycle; FF-MAC; IEEE 802.15.4; NS-3; Test-bed

Laboratoires de recherche :

LIMOS : Informatique, Modélisation et Optimisation des Systèmes

STIC : Sciences et Technologies de l’Information et de la communication